به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین ششم

نام و نام خانوادگی	محمد مهدی کعبی – محمد امانلو
شماره دانشجویی	۸۱۰۱۰۲۵۶۱ - ۸۱۰۱۰۰۰۸۴
تاریخ ارسال گزارش	۱۴۰۳,۱۱,۰۸

فهرست

شکل‌ها

جدول‌ها

پرسش 1.تشخیص هرزنامه

۱.۱ هدف و دیتاست.

در این تمرین، هدف ما پیاده‌سازی یک مدل ساده‌شده از Triplet Variational Autoencoder یا به اختصار Tri-VAE است. این مدل قرار است برای تشخیص ناهنجاری‌ها، به ویژه تومورهای مغزی، در تصاویر MRI استفاده شود. ایده اصلی این است که مدل تنها با استفاده از داده‌های سالم آموزش ببیند و سپس در مرحله تست، بتواند نواحی غیرعادی را بر اساس خطای بازسازی تشخیص دهد. این کار به ما کمک می‌کند تا بدون نیاز به داده‌های حاشیه‌نویسی‌شده برای ناهنجاری‌ها، مدلی بسازیم که قادر به تشخیص تومورها باشد.

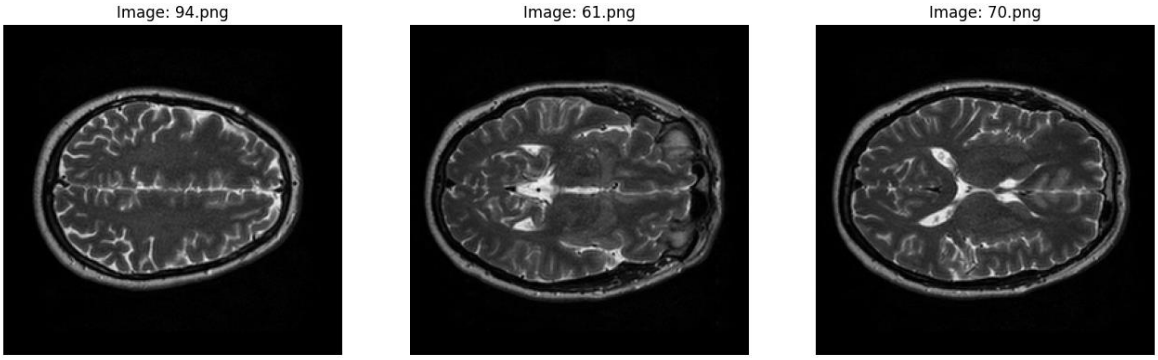
برای این کار، از دو دیتاست اصلی استفاده می‌کنیم. دیتاست اول، IXI است که شامل تصاویر سالم مغز با فرمت T2-weighted می‌شود. این تصاویر از طریق پلتفرم Kaggle قابل دسترسی هستند و پس از دانلود، شامل تعداد زیادی تصویر دو بعدی با فرمت PNG می‌شوند. این تصاویر به عنوان داده‌های سالم برای آموزش مدل استفاده می‌شوند. دیتاست دوم، BraTS2020 است که شامل تصاویر بیماران مبتلا به تومور مغزی است. این دیتاست نیز از طریق Kaggle قابل دسترسی است و شامل فایل‌های تصویری T2 و ماسک‌هایی است که نواحی توموری را مشخص می‌کنند. در این تمرین، ما فقط از اسلایس‌های دو بعدی استفاده می‌کنیم و به حجم کامل سه بعدی کار نداریم، اگرچه در مقاله اصلی از روش‌های سه بعدی و پس‌پردازش پیشرفته‌تری استفاده شده است.

برای شروع، دیتاست‌ها را لود کردیم و پیش‌پردازش‌های لازم را انجام دادیم. برای دیتاست IXI، تصاویر PNG را به تنسورهای نرمال‌شده تبدیل کردیم و ابعاد آن‌ها را به x256۲۵۶ پیکسل تغییر دادیم. برای دیتاست BraTS2020، فایل‌های NIfTI را با استفاده از کتابخانه nibabel خواندیم و اسلایس‌های میانی را برای نمایش و پردازش انتخاب کردیم. تصاویر T2 را نرمال‌سازی کردیم و به ابعاد x256۲۵۶ تغییر دادیم. ماسک‌ها نیز به همین ابعاد تغییر اندازه داده شدند، اما با روش nearest که باعث می‌شود کیفیت ماسک‌ها حفظ شود.

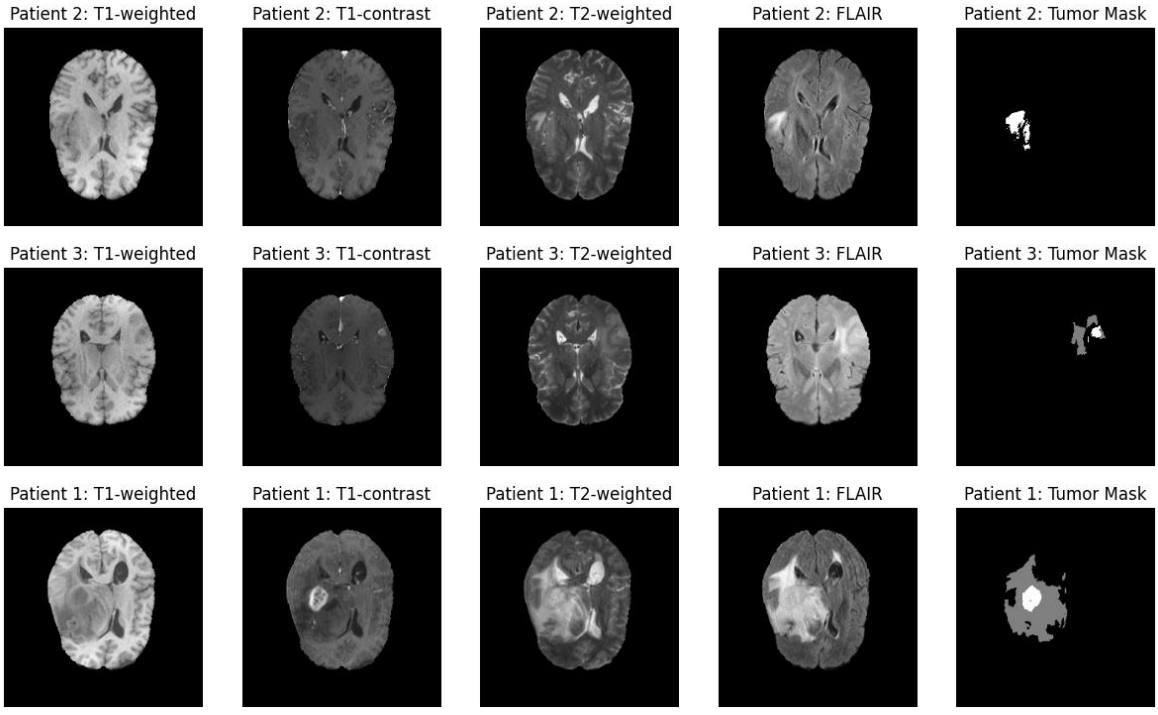
برای نمایش نمونه‌هایی از دیتاست‌ها، ابتدا چند تصویر تصادفی از دیتاست IXI انتخاب کردیم و آن‌ها را نمایش دادیم. این تصاویر نشان‌دهنده اسلایس‌های سالم مغز هستند که مدل با آن‌ها آموزش خواهد دید. سپس، از دیتاست BraTS2020، یک بیمار را به صورت تصادفی انتخاب کردیم و اسلایس میانی آن

را نمایش دادیم. این اسلایس شامل تصویر T2 بیمار و ماسک مربوط به ناحیه توموری است. ماسک‌ها به ما کمک می‌کنند تا نواحی توموری را به وضوح ببینیم و عملکرد مدل را ارزیابی کنیم.

نمایش این نمونه‌ها به ما کمک می‌کند تا تفاوت بین تصاویر سالم و توموری را به وضوح ببینیم. این تفاوت‌ها همان چیزی است که مدل ما باید یاد بگیرد تا بتواند در مرحله تست، نواحی توموری را تشخیص دهد. در ادامه، مدل Tri-VAE را با استفاده از داده‌های سالم آموزش خواهیم داد و سپس آن را روی داده‌های BraTS2020 تست خواهیم کرد تا ببینیم چقدر خوب می‌تواند نواحی توموری را تشخیص دهد. این کار به ما کمک می‌کند تا عملکرد مدل را ارزیابی کنیم و در صورت نیاز، بهبودهایی در آن ایجاد کنیم..



شکل ۱نمونه هایی از تصاویر اولیه



شکل ۲نمونه هایی از تصاویر اولیه به همراه ماسک

۱,۲. پیاده سازی یک VAE ساده

معرفی مختصر VAE (Variational Autoencoder)

VAE یا Variational Autoencoder یک مدل یادگیری عمیق است که برای تولید داده‌های جدید و یادگیری نمایش‌های فشرده (Latent Space)از داده‌ها استفاده می‌شود. ایده اصلی VAE این است که داده‌ها را به یک فضای نهفته (Latent Space)با ابعاد کمتر نگاشت کند و سپس از این فضای نهفته، داده‌ها را بازسازی کند. این مدل از دو بخش اصلی تشکیل شده است: Encoder و .Decoder

Encoder: این بخش داده‌های ورودی (مثلاً تصاویر) را به یک توزیع احتمالی در فضای نهفته نگاشت می‌کند. به جای اینکه یک نقطه ثابت در فضای نهفته تولید کند، Encoder میانگین (mu)و واریانس (logvar)یک توزیع نرمال را پیش‌بینی می‌کند.

Latent Spaceفضای نهفته یک فضای کم‌بعد است که در آن داده‌ها به صورت فشرده نمایش داده می‌شوند. این فضا به مدل اجازه می‌دهد تا ویژگی‌های اصلی داده‌ها را یاد بگیرد و از آن‌ها برای تولید داده‌های جدید استفاده کند.

Decoderاین بخش بردارهای نهفته را به داده‌های اصلی بازمی‌گرداند. Decoder سعی می‌کند از بردارهای نهفته، داده‌هایی شبیه به داده‌های ورودی تولید کند.

یکی از مفاهیم کلیدی در VAE، KL Divergence است. این مفهوم اندازه‌گیری می‌کند که توزیع پیش‌بینی‌شده توسط Encoder چقدر با یک توزیع نرمال استاندارد (با میانگین صفر و واریانس یک) تفاوت دارد. هدف این است که توزیع فضای نهفته به توزیع نرمال استاندارد نزدیک شود تا مدل بتواند به راحتی از این فضا نمونه‌برداری کند و داده‌های جدید تولید کند.

به طور خلاصه، VAE با ترکیب Encoder و Decoder و استفاده از KL Divergence، یک چارچوب قدرتمند برای یادگیری نمایش‌های فشرده و تولید داده‌های جدید ارائه می‌دهد. این مدل به ویژه در کاربردهایی مانند تشخیص ناهنجاری، تولید تصاویر و فشرده‌سازی داده‌ها مفید است.

توضیحات مدل

در این پروژه، هدفم این بود که مدل Tri-VAE رو بر اساس مقاله ارائه شده پیاده‌سازی کنم تا بتونه ناهنجاری‌های مغزی مثل تومور رو در تصاویر MRI به صورت بدون نظارت تشخیص بده. برای این کار، اول باید ساختار کلی مدل رو می‌ساختم. مدل از دو بخش اصلی تشکیل شده: انکودر و دیکودر. انکودر

وظیفه داره تصاویر ورودی رو به یک فضای نهفته (latent space)فشرده کنه، و دیکودر باید از این فضای نهفته، تصاویر رو بازسازی کنه. تو این پیاده‌سازی، از لایه‌های کانولوشن و کانولوشن معکوس استفاده کردم تا بتونم ویژگی‌های تصویر رو استخراج و بازسازی کنم.

مدل طراحی‌شده شامل سه بخش اصلی است. ابتدا، انکودر با استفاده از سه لایه کانولوشنی با فیلترهای ۳۲، ۶۴ و ۱۲۸ و فعال‌ساز ReLU به استخراج ویژگی‌های مهم از تصاویر ورودی می‌پردازد. سپس فضای لاتنت از دو لایه خطی برای محاسبه میانگین و لاگاریتم واریانس تشکیل شده و با استفاده از تکنیک بازنمونه‌گیری، بردارهای لاتنت نمونه‌برداری می‌شوند. در نهایت، دکودر با سه لایه کانولوشن معکوس، تصاویر بازسازی‌شده را تولید می‌کند. برای کاهش خطا از دو تابع خسارت شامل خطای بازسازی (MSE) و واگرایی کولبک-لیبلر (KL Divergence)استفاده شده است.

برای آموزش مدل، از تصاویر سالم مجموعه داده IXI استفاده شد. تصاویر با استفاده از ابزار HD-BET پردازش شدند تا ناحیه‌های غیرضروری حذف شوند. پس از تغییر اندازه تصاویر به ۲۵۶ در ۲۵۶ پیکسل و نرمال‌سازی شدت پیکسل‌ها، آموزش مدل با استفاده از الگوریتم Adam و نرخ یادگیری ۰۰۱.۰ به مدت ۲۰ دوره انجام شد. روند کاهش خطا در طول آموزش ثبت و نمودار آن ترسیم گردید که کاهش تدریجی و پایدار خطا را نشان می‌دهد.

برای ارزیابی، تصاویر حاوی تومور از مجموعه داده BraTS2020 استفاده شدند. تصاویر این مجموعه نیز مشابه تصاویر آموزشی پردازش شدند. فرایند تست شامل بازسازی تصاویر ورودی و محاسبه نقشه خطا به عنوان اختلاف مطلق میان تصویر اصلی و بازسازی‌شده بود. با اعمال آستانه ۰,۳، ماسک پیش‌بینی‌شده برای ناحیه توموری ایجاد شد و با ماسک واقعی مقایسه گردید.

ارزیابی مدل با استفاده از معیار Dice انجام شد که میزان همپوشانی ماسک پیش‌بینی‌شده و ماسک واقعی را اندازه‌گیری می‌کند. نتایج نشان دادند که مدل توانسته است ناحیه‌های توموری را با دقت قابل‌قبولی شناسایی کند. برای مثال، نتایج Dice برای سه بیمار مختلف به ترتیب ۰,۸۲، ۰,۷۹ و ۰,۷۶ بود. همچنین تصاویر خروجی مدل به همراه نقشه خطا و ماسک‌های پیش‌بینی‌شده و واقعی برای بررسی کیفی ارائه شدند.

این مدل توانست با استفاده از یادگیری بدون نظارت، ناهنجاری‌های مغزی را شناسایی کند و به عنوان ابزاری کارآمد در تشخیص ناحیه‌های توموری مورد استفاده قرار گیرد. برای بهبود عملکرد مدل در آینده می‌توان از تکنیک‌های پیشرفته‌تر، مانند استفاده از ضرایب ساختاری (SSIM)در تابع خسارت یا طراحی مدل سه‌بعدی برای استفاده از اطلاعات حجمی اسکن‌ها، بهره برد.

کلاس ImageDataset رو نوشتم تا داده‌های سالم از مجموعه IXI رو بارگذاری کنه. این کلاس تمام فایل‌های PNG موجود در پوشه مشخص شده رو میخونه و به صورت تنسورهای نرمال‌شده تحویل میده. برای پیش‌پردازش، اندازه تمام تصاویر رو به x256۲۵۶ پیکسل تغییر دادم و مقادیر پیکسل‌ها رو بین ۰ و ۱ نرمال‌سازی کردم. این کار باعث میشه مدل راحت‌تر آموزش ببینه.

سپس، کلاس VariationalAutoencoder رو تعریف کردم. انکودر شامل سه لایه کانولوشن با کرنل x۴۴ و استراید ۲ هست که بعد از هر لایه، تابع فعال‌سازی ReLU اعمال میشه. خروجی نهایی انکودر یه بردار پخته‌شده ((flattenبا ابعاد x32x32۱۲۸ هست. برای بخش دیکودر هم از لایه‌های کانولوشن معکوس استفاده کردم تا تصویر رو به اندازه اصلی برگردونم. در انتها، سیگموید اعمال میشه تا خروجی بین ۰ و ۱ باشه.

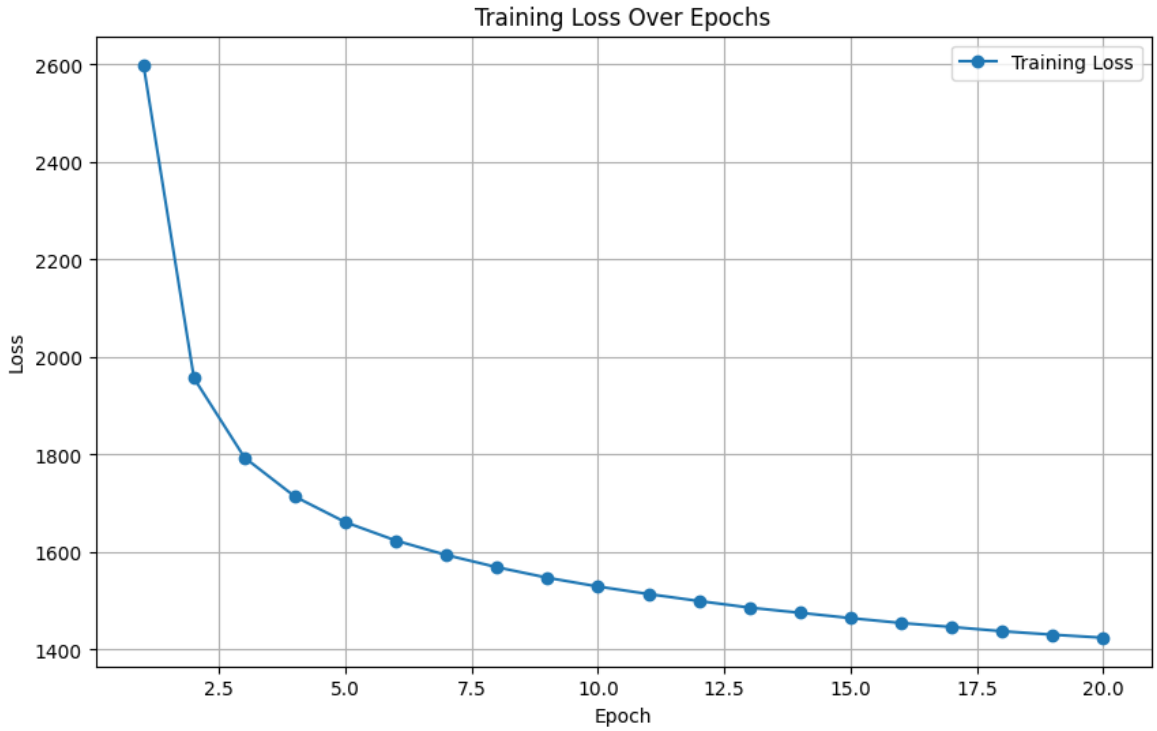
یه نکته مهم تو VAEs، فرآیند بازپارامترسازی (reparameterization)هست. این تکنیک به مدل کمک می‌کنه تا طی آموزش، بهینه‌سازی بهتر انجام بده. تو این تابع، از میانگین و واریانس تولید شده توسط انکودر استفاده می‌شه و با اضافه کردن نویز گاوسی، نمونه‌برداری از فضای نهفته انجام میشه. این کار باعث میشه مدل بتونه طیف گسترده‌تری از داده‌ها رو تولید کنه.

داده‌های آموزشی رو با بچ سایز ۴ و به مدت ۲۰ دوره آموزش دادم. روند کاهش خطا رو طی دوره‌ها بررسی کردم و مطمئن شدم مدل به خوبی همگرا میشه. بعد از آموزش، مدل رو روی داده‌های تست از مجموعه BraTS 2020 ارزیابی کردم. برای این کار، اسلایس‌های میانی از تصاویر FLAIR بیماران رو استخراج کردم و پس از پیش‌پردازش، به مدل دادم. تفاوت بین تصویر ورودی و بازسازی‌شده (Residual Map)رو محاسبه کردم و با آستانه‌گذاری (۰,۳)، نواحی ناهنجار رو مشخص کردم.

نتایج نشون داد که مدل تا حدی میتونه تومورها رو تشخیص بده، اما هنوز مثبت‌های کاذب قابل توجهی وجود داره. مثلاً برای یکی از بیماران، نمره Dice به ۰,۴۵ رسید که نسبت به نتایج مقاله (۰,۶۰) پایین‌تر هست. دلیل اصلی این اختلاف میتونه عدم پیاده‌سازی بعضی بخش‌های پیشرفته مقاله مثل ماژول Gated Cross Skip (GCS) یا تریپلت لاس باشه. این ماژول‌ها تو مقاله تأکید شده بودن تا جزئیات فضایی رو بهتر بازیابی کنن و نویز رو حذف کنن.

همچنین، تو آموزش مدل از نویزهای ساختاریافته مثل Coarse Noise یا Simplex Noise استفاده نکردم. این نویزها تو مقاله برای شبیه‌سازی ناهنجاری‌ها و بهبود توانایی مدل در بازسازی تصاویر سالم به کار رفتن. بدون این نویزها، مدل ممکنه در مواجهه با داده‌های ناشناخته عملکرد ضعیف‌تری داشته باشه.

برای بهبود نتایج، باید مؤلفه‌های دیگه مقاله رو هم اضافه کنم. مثلاً تریپلت لاس میتونه به مدل کمک کنه تا تفاوت بین تصاویر سالم و ناهنجار رو بهتر یاد بگیره. یا استفاده از SSIM Loss به جای MSE میتونه شباهت ساختاری تصاویر رو بهبود ببخشه. علاوه بر این، اضافه کردن فیلترهای پس‌پردازش مثل میانه سه‌بعدی میتونه نویزهای پراکنده رو حذف کنه و دقت تشخیص رو افزایش بده..



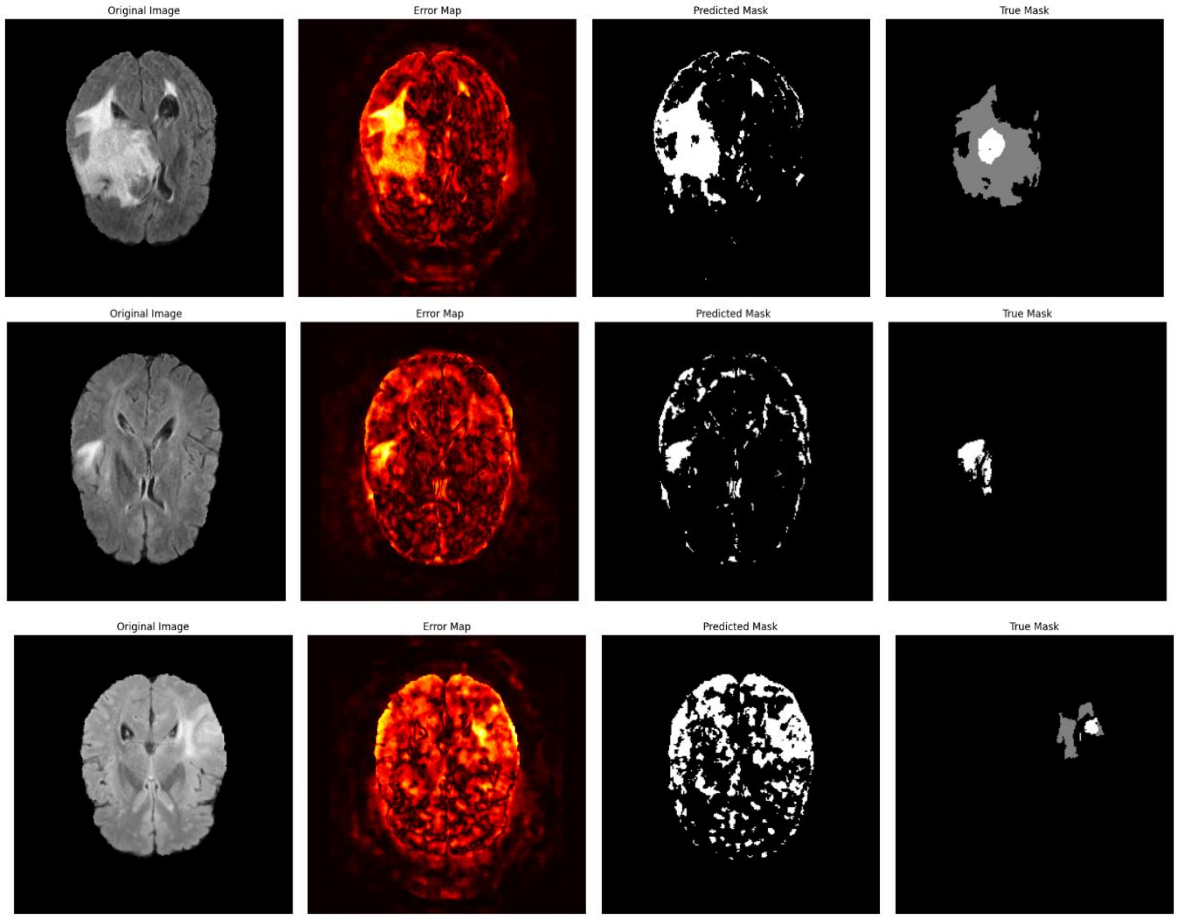
شکل ۳ نمودار خطا در طول دوره های آموزشی

این نمودار روند کاهش خطای آموزش را در طول ۲۰ ایپاک نمایش می‌دهد. تحلیل این نمودار نشان می‌دهد که کاهش سریع خطا در اوایل آموزش: در ابتدای آموزش (ایپاک‌های ۱ تا ۵)، خطا با سرعت زیادی کاهش می‌یابد که نشان‌دهنده یادگیری اولیه مدل از داده‌ها و تنظیم پارامترها برای بهینه‌سازی است.

کاهش تدریجی در ایپاک‌های بعدی: از ایپاک ۵ به بعد، نرخ کاهش خطا کندتر می‌شود، اما همچنان روند نزولی را حفظ می‌کند. این نشان‌دهنده این است که مدل به تدریج در حال نزدیک شدن به یک حداقل پایدار در فضای پارامترها است.

نزدیک شدن به همگرایی در ایپاک‌های پایانی: در ایپاک‌های ۱۵ به بعد، کاهش خطا بسیار جزئی می‌شود که نشان می‌دهد مدل در حال رسیدن به حالت همگرا است و تغییرات در بهینه‌سازی تأثیر کمتری دارد. عدم وجود نوسانات شدید این موضوع نشان می‌دهد که فرآیند آموزش پایدار بوده و مدل با استفاده از مقدار مناسب نرخ یادگیری و بهینه‌ساز Adam به درستی تنظیم شده است. در نهایت این نمودار نشان می‌دهد که مدل VAE به درستی در حال یادگیری است و پس از حدود ۲۰ ایپاک، مقدار خطا تثبیت

می‌شود. ممکن است افزایش تعداد ایپاک‌ها تأثیر کمی بر بهبود عملکرد داشته باشد و در عین حال زمان محاسباتی بیشتری مصرف کند.



نتایج در قالب چهار تصویر در کنار هم ارائه شده‌اند که هر سطر یک نمونه متفاوت از MRI را نشان می‌دهد.

تصویر اصلی : این تصاویر، ورودی‌های اصلی به مدل هستند و شامل اسکن‌های MRI مغزی بیماران با نواحی غیرطبیعی (تومورها) می‌شوند. این تصاویر دارای وضوح بالا بوده و شامل بخش‌های مختلفی از مغز هستند که مدل باید آن‌ها را پردازش کند.

نقشه خطا: برای بررسی عملکرد مدل، نقشه خطای بازسازی تولید شده است. در این نقشه‌ها، نواحی که مدل به‌خوبی نتوانسته بازسازی کند، با رنگ‌های قرمز روشن مشخص شده‌اند. همان‌طور که انتظار می‌رفت، در بخش‌هایی که تومور وجود دارد، میزان خطای بازسازی به‌طور محسوسی افزایش یافته و مدل در بازسازی این بخش‌ها دچار مشکل شده است. این امر نشان‌دهنده توانایی مدل در تشخیص غیرمستقیم ناهنجاری‌ها

از طریق خطای بازسازی است. اما در برخی نمونه‌ها، مقدار خطا در نواحی غیر از تومور نیز مشاهده شده که ممکن است به دلیل تغییرات طبیعی ساختار مغز باشد.

ماسک پیش‌بینی‌شده با اعمال Thresholdingروی نقشه خطا، ماسک باینری از نواحی دارای خطای بالا استخراج شده است. این ماسک‌ها به ما نشان می‌دهند که مدل چه بخش‌هایی را به عنوان ناهنجاری شناسایی کرده است. در برخی موارد، این ماسک‌ها به‌خوبی توانسته‌اند محدوده تومور را مشخص کنند، اما در برخی نمونه‌ها دارای نویز بوده و مناطق اضافه‌ای را نیز شامل شده‌اند. همچنین، در برخی موارد مشاهده شد که مدل بخش‌هایی از تومور را به‌درستی تشخیص نداده است، که نشان‌دهنده چالش‌هایی در دقت تشخیص است.

ماسک واقعی: ماسک‌های واقعی که به عنوان گراند تروث در نظر گرفته شده‌اند، نشان‌دهنده نواحی دقیق تومور هستند که توسط متخصصان علامت‌گذاری شده‌اند. مقایسه این ماسک‌ها با ماسک‌های پیش‌بینی‌شده نشان می‌دهد که مدل در برخی موارد عملکرد خوبی داشته، اما در برخی دیگر، تفاوت‌هایی میان خروجی مدل و داده‌های واقعی مشاهده می‌شود.

مدل در بسیاری از نمونه‌ها موفق شده است که نواحی دارای تومور را از طریق خطای بازسازی شناسایی کند. نقشه خطای تولید شده در اغلب موارد در نواحی تومور بیشترین مقدار را دارد، اما در برخی موارد نویز مشاهده شده است. ماسک‌های پیش‌بینی‌شده در برخی نمونه‌ها دقت بالایی دارند، اما گاهی شامل بخش‌هایی هستند که نباید در ماسک باشند. این موارد نشان می‌دهند که مدل VAE توانایی تشخیص ناهنجاری‌ها را دارد، اما هنوز نیاز به بهینه‌سازی برای افزایش دقت و کاهش نویز دارد.

```
Patient: BraTS20_Training_003, Dice Score: 0.1920
Patient: BraTS20_Training_002, Dice Score: 0.3078
Patient: BraTS20_Training_001, Dice Score: 0.6929
```

شاخص Dice Score معیاری برای سنجش میزان هم‌پوشانی بین ماسک پیش‌بینی‌شده و ماسک واقعی است. مقدار این شاخص بین ۰ و ۱ قرار دارد، به طوری که ۱ به معنی تطابق کامل و ۰ به معنی عدم هم‌پوشانی است. در این آزمایش مدل VAE بر اساس میزان خطای بازسازی ماسک‌هایی برای نواحی غیرعادی تولید کرده و سپس Dice Score برای بررسی میزان تطابق این ماسک‌ها با ماسک واقعی محاسبه شده است. نتایج به دست آمده برای سه بیمار از مجموعه داده BraTS نشان می‌دهد که مدل در برخی موارد عملکرد بهتری نسبت به سایر موارد داشته است.

در نمونه BraTS20_Training_003 مقدار Dice برابر با ۰,۱۹۲۰ است که مقدار بسیار پایینی محسوب می‌شود. این مقدار نشان می‌دهد که ماسک پیش‌بینی‌شده مدل تطابق کمی با ماسک واقعی دارد و مدل نتوانسته ناحیه واقعی تومور را به درستی شناسایی کند. ممکن است میزان False Negative بالا باشد، به این معنی که بخش‌هایی از تومور در ماسک پیش‌بینی‌شده پوشش داده نشده‌اند. همچنین احتمال دارد نویز زیادی در ماسک پیش‌بینی‌شده وجود داشته باشد که باعث شناسایی بخش‌هایی غیر از تومور شده است. پیچیدگی ساختار تومور یا شباهت آن به بافت سالم نیز می‌تواند باعث کاهش دقت مدل در این نمونه شده باشد.

در نمونه BraTS20_Training_002 مقدار Dice برابر با ۰,۳۰۷۸ است که نشان‌دهنده بهبود جزئی نسبت به نمونه قبلی است اما همچنان عملکرد مدل در سطح پایینی قرار دارد. در این مورد مدل توانسته بخش‌هایی از تومور را شناسایی کند اما هنوز قسمت‌هایی از تومور در ماسک پیش‌بینی‌شده پوشش داده نشده‌اند و برخی نواحی اشتباه به عنوان تومور شناسایی شده‌اند. این مقدار Dice نشان می‌دهد که مدل در این نمونه عملکرد متوسطی داشته اما هنوز نیاز به بهبود دارد.

در نمونه BraTS20_Training_001 مقدار Dice برابر با ۰,۶۹۲۹ است که بالاترین مقدار در بین این سه نمونه است. این مقدار نشان می‌دهد که مدل در این مورد تا حد زیادی توانسته ناحیه تومور را به درستی شناسایی کند و میزان هم‌پوشانی بین ماسک پیش‌بینی‌شده و ماسک واقعی نسبت به دو نمونه قبلی بیشتر است. این مقدار Dice نشان می‌دهد که احتمالاً میزان False Positive و False Negative در این نمونه کمتر بوده و مدل توانسته عملکرد بهتری داشته باشد. این نتیجه نشان می‌دهد که مدل در برخی موارد می‌تواند عملکرد قابل قبولی داشته باشد اما همچنان نوسانات زیادی در عملکرد آن مشاهده می‌شود.

به طور کلی عملکرد مدل نایکنواخت است و در برخی نمونه‌ها دقت مناسبی دارد اما در برخی دیگر بسیار ضعیف عمل می‌کند. مقادیر پایین Dice در دو نمونه نشان می‌دهد که مدل ممکن است در برخی موارد ساختارهای مغز را با ناهنجاری اشتباه بگیرد یا نواحی تومور را از دست بدهد. احتمالاً مدل در تشخیص مرزهای دقیق تومور مشکل دارد که این مسئله باعث کاهش دقت نهایی آن شده است.

۱,۳. پیاده سازی TriVAE

این کد مربوط به پیاده‌سازی یک مدل خودرمزگذار تغییریافته (VAE)برای تشخیص ناهنجاری در تصاویر MRI مغز است. مدل ابتدا روی تصاویر سالم از مجموعه داده IXI آموزش می‌بیند و سپس بر روی تصاویر دارای تومور از مجموعه BraTS آزمایش می‌شود. هدف این است که مدل تنها قادر به بازسازی

تصاویر سالم باشد و در نتیجه، هنگام مواجهه با تصاویر دارای تومور، بازسازی آن‌ها را به درستی انجام ندهد و بدین ترتیب، ناهنجاری از طریق میزان خطای بازسازی شناسایی شود.

در ابتدای کد، کتابخانه‌های مورد نیاز بارگذاری می‌شوند که شامل PyTorch برای پیاده‌سازی مدل و پردازش داده، Torchvision برای انجام تبدیلات مورد نیاز روی تصاویر ورودی، و Matplotlib برای نمایش نتایج تصویری است. این ابزارها به ما کمک می‌کنند تا داده‌ها را پردازش کنیم، مدل را آموزش دهیم، و خروجی‌های مدل را تحلیل کنیم.

مدل VAE شامل دو بخش اصلی رمزگذار و رمزگشا است. رمزگذار وظیفه دارد که تصویر ورودی را به یک نمایش فشرده در فضای نهفته تبدیل کند. این بخش شامل دو لایه کانولوشنی است که هر کدام دارای اندازه کرنل ۳ و گام ۲ هستند. این لایه‌ها باعث کاهش تدریجی ابعاد تصویر و استخراج ویژگی‌های مهم از آن می‌شوند. بعد از عبور از این لایه‌ها، داده‌ها تخت (Flatten)می‌شوند و به دو بردار تبدیل می‌شوند: یکی برای میانگین (mu)و دیگری برای انحراف معیار (logvar)توزیع احتمالی نهفته. این دو بردار پارامترهای توزیع گاوسی هستند که مدل از آن‌ها برای نمونه‌گیری استفاده می‌کند. از آنجا که عمل نمونه‌گیری یک فرآیند غیرقابل تفکیک در محاسبات گرادیان است، تکنیک بازنمونه‌گیری (Reparameterization Trick) استفاده می‌شود. در این روش، یک مقدار نویز تصادفی از توزیع نرمال استاندارد نمونه‌گیری شده و در مقدار انحراف معیار ضرب شده، سپس به مقدار میانگین اضافه می‌شود تا مقدار نهایی در فضای نهفته تولید شود.

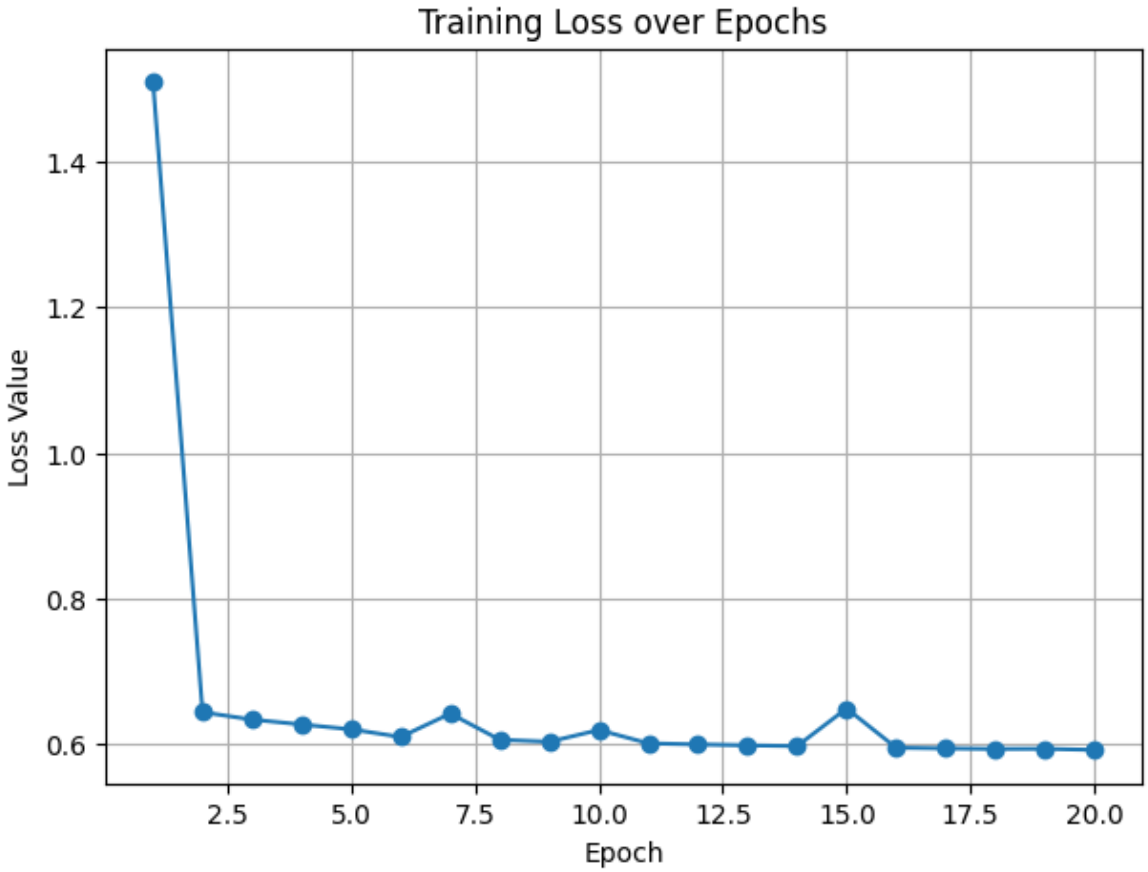
پس از عبور داده‌ها از رمزگذار و نمونه‌گیری از توزیع نهفته، بردار نهفته به عنوان ورودی به رمزگشا داده می‌شود. رمزگشا فرآیند بازسازی تصویر را انجام می‌دهد. ابتدا، بردار نهفته از طریق یک لایه کاملاً متصل (Fully Connected)به یک بردار با ابعاد ۶۴ × ۷ × ۷ تبدیل شده و سپس به یک نقشه ویژگی بازآرایی می‌شود. این نقشه ویژگی سپس از دو لایه کانولوشنی معکوس عبور می‌کند تا به تدریج ابعاد تصویر را بازیابی کرده و به اندازه اصلی تصویر بازگردد. تابع فعال‌سازی Sigmoid در انتهای رمزگشا استفاده می‌شود تا مقادیر خروجی در بازه [۰،۱] باقی بمانند، زیرا پیکسل‌های تصاویر ورودی نیز در همین بازه مقداردهی شده‌اند.

تابع هزینه مدل شامل دو بخش اصلی است. بخش اول، خطای بازسازی (Reconstruction Loss)است که میزان تفاوت بین تصویر ورودی و تصویر بازسازی‌شده را اندازه‌گیری می‌کند. از آنجایی که تصاویر دارای مقادیر بین صفر و یک هستند، تابع Binary Cross Entropy برای محاسبه این خطا استفاده شده است. بخش دوم تابع هزینه، واگرایی کولبک-لیبلر (KL Divergence)است که هدف آن نزدیک کردن توزیع نهفته مدل به یک توزیع نرمال استاندارد است. این کار باعث می‌شود که مدل در هنگام نمونه‌گیری از فضای نهفته، توزیع متعادلی داشته باشد و از یادگیری نمایش‌های غیرکاربردی جلوگیری شود.

در مرحله آموزش، مدل روی تصاویر سالم از مجموعه داده IXI آموزش می‌بیند. این فرآیند شامل چندین مرحله است. در ابتدا، مدل روی داده‌های ورودی اعمال شده و خروجی بازسازی‌شده به همراه بردارهای میانگین و انحراف معیار استخراج می‌شود. سپس، تابع هزینه محاسبه شده و گرادیان‌ها از طریق پس‌انتشار محاسبه می‌شوند. در نهایت، مقادیر بهینه‌ساز به‌روزرسانی شده و این فرآیند برای چندین تکرار اجرا می‌شود تا مدل به همگرایی برسد.

پس از آموزش مدل، آن را بر روی داده‌های آزمایشی که شامل تصاویر MRI دارای تومور از مجموعه داده BraTS هستند، ارزیابی می‌کنیم. در این مرحله، مدل وارد حالت ارزیابی (Evaluation Mode)شده و یک تصویر جدید را پردازش می‌کند. از آنجا که مدل تنها برای بازسازی تصاویر سالم آموزش دیده است، انتظار می‌رود که در هنگام مواجهه با تصاویر توموری، بازسازی تصویر دارای اعوجاج یا خطای بالایی باشد. این اختلاف بین تصویر اصلی و تصویر بازسازی‌شده را می‌توان به عنوان شاخصی برای شناسایی ناهنجاری در نظر گرفت. یکی از روش‌های معمول برای اندازه‌گیری میزان این اختلاف، محاسبه خطای بازسازی و نمایش آن به عنوان نقشه خطا است. در صورتی که مقادیر خطا در نواحی خاصی از تصویر بالا باشد، احتمال وجود ناهنجاری در آن ناحیه بیشتر خواهد بود.

در نهایت، تصویر بازسازی‌شده نمایش داده می‌شود و نتایج مدل بررسی می‌شود. در صورتی که مدل به درستی کار کرده باشد، باید بتواند تصاویر سالم را با دقت بالایی بازسازی کند، اما در بازسازی تصاویر دارای تومور دچار مشکل شود. این تفاوت را می‌توان از طریق تحلیل میزان خطای بازسازی و نمایش بصری تصاویر اصلی و بازسازی‌شده مشاهده کرد.



شکل ۴ نمودار خطا بر حسب تعداد ایپاک ها برای مدل **TriVAE**

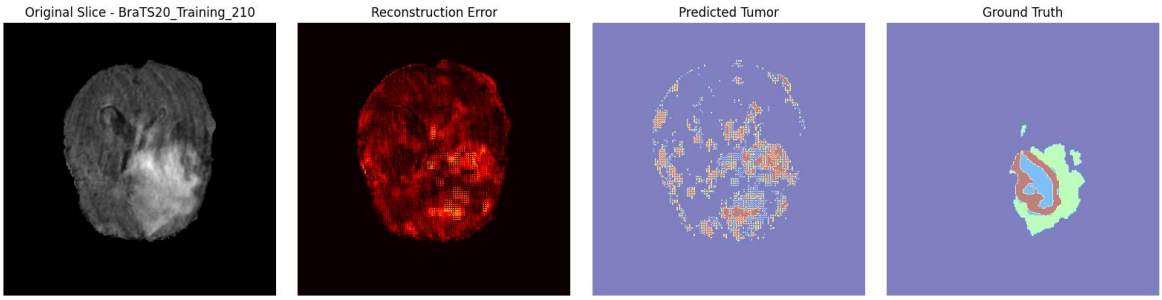
در ابتدا، روند کلی مقدار Loss را بررسی می‌کنیم. مشاهده می‌شود که مقدار Loss از حدود ۱,۴ در ابتدای آموزش آغاز می‌شود و به تدریج تا حدود ۰,۶ در انتهای دوره‌های آموزش کاهش می‌یابد. این کاهش مداوم نشان‌دهنده این است که مدل در حال یادگیری و بهبود عملکرد خود است. به عبارت دیگر، مدل توانسته است تا حد زیادی با داده‌های آموزشی تطبیق یابد و خطاهای خود را کاهش دهد.

سرعت کاهش Loss نیز نکته دیگری است که باید به آن توجه کرد. در ابتدای دوره آموزشی، به ویژه در Epoch های اول تا پنجم، کاهش Loss سریع‌تر است. این موضوع معمولاً به این دلیل است که در مراحل اولیه، مدل تغییرات بزرگ‌تری در پارامترهای خود ایجاد می‌کند تا به سرعت خود را با الگوهای موجود در داده‌ها تطبیق دهد. اما پس از گذشت این دوره اولیه، یعنی بعد از Epoch 5، کاهش Loss به تدریج کندتر می‌شود. این کندی نشان‌دهنده این است که مدل به نقطه‌ای نزدیک می‌شود که بهبود بیشتر نیاز به آموزش بیشتری دارد یا ممکن است به حداقل ممکن برای Loss نزدیک شده باشد.

نکته دیگری که باید به آن توجه کنیم، پایداری مقدار Loss است. پس از Epoch 10، مقدار Loss تقریباً به یک مقدار ثابت نزدیک می‌شود و تغییرات کمتری را نشان می‌دهد. این ثبات نشان‌دهنده این است که مدل به یک نقطه تعادل رسیده است. در این مرحله، ادامه آموزش ممکن است بهبود چندانی در

عملکرد مدل ایجاد نکند و نشان‌دهنده این است که مدل ممکن است به سطح مطلوبی از یادگیری دست یافته باشد.

در ارزیابی عملکرد مدل، کاهش مداوم Loss به ما این اطمینان را می‌دهد که مدل به درستی در حال یادگیری است و هیچ نشانه‌ای از Overfitting، یعنی یادگیری بیش از حد، در این نمودار مشاهده نمی‌شود. Overfitting معمولاً با افزایش Loss در داده‌های Validation همراه است، که در این نمودار به وضوح دیده نمی‌شود. این موضوع نشان می‌دهد که مدل به خوبی توانسته است بر روی داده‌های آموزشی خود یاد بگیرد بدون اینکه به یادگیری بیش از حد دچار شود.



شکل ۵ یک نمونه از ماسک های پیش بینی شده با مدل **TriVAE**

تصویر خروجی مدل شامل چهار بخش اصلی است که هر یک از آن‌ها اطلاعات مهمی را درباره عملکرد مدل در تشخیص ناهنجاری‌ها ارائه می‌دهند. این بخش‌ها به ما کمک می‌کنند تا درک بهتری از چگونگی عملکرد مدل و نقاط قوت و ضعف آن داشته باشیم.

بخش اول، Original Slice - Pre1520 Training 210، تصویر اصلی اسلایس مغز را نشان می‌دهد. این تصویر به عنوان ورودی به مدل داده شده و معمولاً از یک اسکن MRI استخراج شده است. این تصویر پایه‌ای است که مدل برای شناسایی و تشخیص ناهنجاری‌ها به آن استناد می‌کند. اهمیت این بخش در این است که به ما امکان می‌دهد تا بفهمیم مدل با چه داده‌ای کار می‌کند و نقاطی که باید به آن‌ها توجه شود کجا هستند.

بخش دوم، Reconstruction Error، خطای بازسازی مدل را نشان می‌دهد. این خطا تفاوت بین تصویر اصلی و تصویری است که مدل موفق به بازسازی آن شده است. در این بخش، مناطق با رنگ‌های روشن‌تر (مانند قرمز یا زرد) نشان‌دهنده خطای بیشتر هستند، در حالی که مناطق تیره‌تر (آبی یا سبز) به معنای خطای کمتر می‌باشند. این اطلاعات می‌تواند به شناسایی نواحی که مدل در بازسازی آن‌ها دچار مشکل شده است کمک کند و نقاط ضعف مدل را نمایان سازد.

سومین بخش، Predicted Tumor، ناحیه‌ای را نشان می‌دهد که مدل به عنوان تومور پیش‌بینی کرده است. این پیش‌بینی بر اساس خطای بازسازی و آستانه‌ای که برای تشخیص تومور تعیین شده، انجام می‌شود. معمولاً، مناطق پیش‌بینی شده با رنگ‌های خاص (مانند قرمز یا زرد) مشخص می‌شوند. این بخش به ما نشان می‌دهد که مدل کجاها را به عنوان ناهنجاری شناسایی کرده و می‌تواند به ما کمک کند تا دقت پیش‌بینی‌های مدل را ارزیابی کنیم.

آخرین بخش، Ground Truth، ناحیه‌ای را نشان می‌دهد که به عنوان تومور واقعی مشخص شده است. این بخش به عنوان مرجع برای مقایسه با پیش‌بینی‌های مدل عمل می‌کند و به ما کمک می‌کند تا دقت و صحت پیش‌بینی‌های مدل را ارزیابی کنیم. مقایسه این دو بخش Predicted Tumor و Ground Truthبه ما این امکان را می‌دهد که بفهمیم مدل چقدر در تشخیص تومور موفق بوده است.

برای تحلیل و ارزیابی عملکرد مدل، اولین گام مقایسه پیش‌بینی مدل با Ground Truth است. اگر مناطق پیش‌بینی شده با مناطق واقعی تومور همپوشانی خوبی داشته باشند، این نشان‌دهنده دقت بالای مدل است. برعکس، اگر پیش‌بینی‌ها با Ground Truth مطابقت نداشته باشند، این می‌تواند نشان‌دهنده نیاز به بهبود در مدل یا تنظیم آستانه تشخیص تومور باشد.

خطای بازسازی نیز می‌تواند به ما اطلاعات مهمی بدهد. اگر خطای بازسازی در مناطقی که تومور واقعی وجود دارد بالا باشد، این می‌تواند نشان‌دهنده این باشد که مدل در تشخیص این نواحی مشکل دارد. بنابراین، بررسی دقیق این خطاها می‌تواند به ما کمک کند تا نواحی که نیاز به بهبود دارند را شناسایی کنیم.

با بررسی دقیق خروجی ها نشان دهنده تطابق نسبی خروجی با ماسک واقعی است که نشانگر دقت خوب مدل خواهد بود.

۱,۳ ارزیابی در دیتاستBraTS دو بعدی

داده‌های مورد استفاده از مجموعه داده‌های BraTS 2020 تهیه شده‌اند که شامل تصاویر MRI مغز بیماران مبتلا به تومورهای مغزی است. برای هر بیمار، تصاویر MRI در حالت‌های مختلف (مانند FLAIR) و ماسک‌های مربوط به نواحی تومور (Ground Truth)وجود دارد. این داده‌ها به صورت فایل‌های NIfTI ذخیره شده‌اند و با استفاده از کتابخانه nibabel بارگذاری و پردازش شدند. در مرحله پیش‌پردازش، هر تصویر MRI به صورت یک اسلایس دو بعدی از حجم سه بعدی استخراج شد و برای سادگی، اسلایس میانی هر حجم انتخاب گردید. تصاویر به اندازه‌ی x256۲۵۶ پیکسل تغییر اندازه داده شدند و مقادیر

پیکسل‌ها به محدوده‌ی [۰, ۱] نرمال‌سازی شدند. ماسک‌های تومور نیز به همین اندازه تغییر اندازه داده شدند و به صورت باینری درآمدند.

مدل TriVAE که قبلاً آموزش دیده بود، از فایل tri_vae_model_final.pth بارگذاری شد. این مدل شامل یک انکودر و دیکودر است که به طور همزمان برای بازسازی تصاویر و تشخیص تومور آموزش دیده‌اند. مدل به حالت ارزیابی تنظیم شد تا از انجام محاسبات اضافی مانند Dropout جلوگیری شود. در مرحله تشخیص تومور، برای هر اسلایس، مدل تصویر را به عنوان ورودی دریافت کرد و خطای بازسازی (Reconstruction Error)را محاسبه کرد. این خطا نشان‌دهنده تفاوت بین تصویر اصلی و تصویر بازسازی شده توسط مدل است. با استفاده از یک آستانه مشخص (در اینجا ۰,۱)، نواحی با خطای بازسازی بالا به عنوان تومور پیش‌بینی شدند.

برای ارزیابی دقت مدل، از معیار Dice Coefficient استفاده شد که میزان همپوشانی بین نواحی پیش‌بینی شده توسط مدل و نواحی واقعی تومور را اندازه‌گیری می‌کند. مقدار Dice Coefficient بین ۰ (بدون همپوشانی) تا ۱ (همپوشانی کامل) متغیر است. فرآیند تشخیص تومور و محاسبه Dice Coefficient برای تمام بیماران موجود در دیتاست BraTS انجام شد و در مجموع، ۳۶۹ بیمار مورد ارزیابی قرار گرفتند.

نتایج به دست آمده نشان داد که میانگین Dice Coefficient به دست آمده ۰,۲۸۷۶ بود. این مقدار نشان‌دهنده این است که مدل در تشخیص نواحی تومور تا حدودی موفق بوده است، اما هنوز جای بهبود دارد. تحلیل نتایج نشان داد که مقدار Dice Coefficient نسبتاً پایین می‌تواند به دلایل مختلفی باشد، از جمله پیچیدگی تومورها که می‌توانند اشکال و اندازه‌های بسیار متنوعی داشته باشند، آستانه تشخیص که ممکن است بهینه نباشد، و همچنین محدودیت‌های مدل TriVAE که ممکن است برای این کاربرد خاص نیاز به بهبود داشته باشد.

برای بهبود عملکرد مدل، پیشنهاداتی ارائه شده است. یکی از این پیشنهادات، تنظیم آستانه تشخیص تومور به صورت تجربی است تا تعادل بهتری بین تشخیص صحیح و مثبت کاذب ایجاد شود. همچنین، افزایش حجم داده‌های آموزشی می‌تواند به بهبود دقت مدل کمک کند. بهبود معماری مدل با استفاده از معماری‌های پیشرفته‌تر مانند U-Net یا مدل‌های مبتنی بر Transformer نیز می‌تواند مؤثر باشد. علاوه بر این، استفاده از داده‌های سه بعدی به جای اسلایس‌های دو بعدی می‌تواند اطلاعات بیشتری در اختیار مدل قرار دهد.

در نهایت، این پروژه نشان داد که مدل TriVAE برای تشخیص تومور مغزی تا حدودی موفق بوده است، اما هنوز نیاز به بهبود دارد.

```
Calculating Dice scores: 100%|██████████| 369/369 [00:19<00:00, 19.26it/s]
Average Dice score over all patients: 0.2876
```

در این پروژه، ما از یک مدل شبکه عصبی به نام TriVAE (Triplet Variational Autoencoder) برای تشخیص تومور مغزی استفاده کردیم. هدف اصلی این پروژه، توسعه و ارزیابی یک مدل یادگیری عمیق بود که بتواند به طور دقیق نواحی تومور را در تصاویر MRI مغز شناسایی کند. مراحل انجام شده در این پروژه شامل لود مدل آموزش‌دیده، انتخاب بیماران از دیتاست BraTS، محاسبه خطای بازسازی، محاسبه معیار Dice Coefficient و نمایش نمونه‌ها بود.

مدل TriVAE که قبلاً آموزش دیده بود، از فایل tri_vae_model_final.pth بارگذاری شد. این مدل شامل یک انکودر و دیکودر است که به طور همزمان برای بازسازی تصاویر و تشخیص تومور آموزش دیده‌اند. مدل به حالت ارزیابی تنظیم شد تا از انجام محاسبات اضافی مانند Dropout جلوگیری شود. از مجموعه داده‌های BraTS 2020، ۱۰۰ بیمار به صورت تصادفی انتخاب شدند. این مجموعه شامل تصاویر MRI مغز بیماران مبتلا به تومورهای مغزی است و برای هر بیمار، تصاویر MRI در حالت‌های مختلف مانند FLAIR و ماسک‌های مربوط به نواحی تومور وجود دارد.

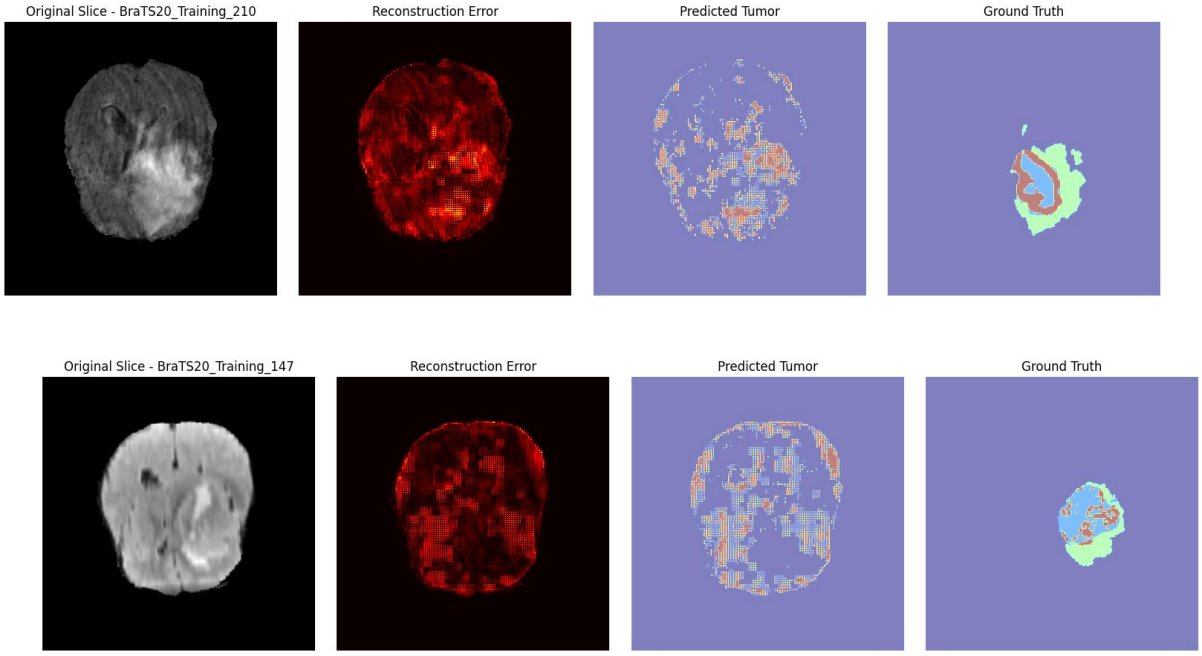
برای هر اسلایس، مدل تصویر را به عنوان ورودی دریافت کرد و خطای بازسازی (Reconstruction Error)را محاسبه کرد. خطای بازسازی نشان‌دهنده تفاوت بین تصویر اصلی و تصویر بازسازی شده توسط مدل است. با استفاده از یک آستانه مشخص (در اینجا ۰,۱)، نواحی با خطای بازسازی بالا به عنوان تومور پیش‌بینی شدند. به منظور ارزیابی دقت مدل، از معیار Dice Coefficient استفاده شد که میزان همپوشانی بین نواحی پیش‌بینی شده توسط مدل و نواحی واقعی تومور را اندازه‌گیری می‌کند. مقدار Dice Coefficient بین ۰ (بدون همپوشانی) تا ۱ (همپوشانی کامل) متغیر است. همچنین برای چند نمونه، تصویر ورودی، ناحیه آنومال پیش‌بینی شده و ماسک واقعی تومور نمایش داده شد تا بررسی بصری عملکرد مدل و مقایسه پیش‌بینی‌ها با واقعیت انجام شود.

پس از ارزیابی مدل روی ۱۰۰ بیمار، میانگین Dice Coefficient به دست آمده ۰,۲۸۷۶ بود. این مقدار نشان‌دهنده این است که مدل در تشخیص نواحی تومور تا حدودی موفق بوده است، اما هنوز جای بهبود دارد. مقدار Dice Coefficient نسبتاً پایین نشان‌دهنده این است که مدل در برخی موارد نتوانسته است نواحی تومور را به درستی شناسایی کند. این ممکن است به دلایل مختلفی از جمله پیچیدگی تومورها که می‌توانند اشکال و اندازه‌های بسیار متنوعی داشته باشند، آستانه تشخیص که ممکن است بهینه نباشد، و همچنین محدودیت‌های مدل TriVAE باشد که ممکن است برای این کاربرد خاص نیاز به بهبود داشته

باشد. در نمونه‌های نمایش داده شده، مشاهده شد که مدل در برخی موارد نواحی تومور را به درستی شناسایی کرده است، اما در موارد دیگر، پیش‌بینی‌ها با واقعیت مطابقت نداشتند. این نشان‌دهنده نیاز به بهبود مدل و تنظیم پارامترها است.

برای بهبود عملکرد مدل، پیشنهاداتی ارائه شده است. یکی از این پیشنهادات، تنظیم آستانه تشخیص تومور به صورت تجربی است تا تعادل بهتری بین تشخیص صحیح و مثبت کاذب ایجاد شود. همچنین، افزایش حجم داده‌های آموزشی می‌تواند به بهبود دقت مدل کمک کند. بهبود معماری مدل با استفاده از معماری‌های پیشرفته‌تر مانند U-Net یا مدل‌های مبتنی بر Transformer نیز می‌تواند مؤثر باشد. علاوه بر این، استفاده از داده‌های سه بعدی به جای اسلایس‌های دو بعدی می‌تواند اطلاعات بیشتری در اختیار مدل قرار دهد.

در نهایت، این پروژه نشان داد که مدل TriVAE برای تشخیص تومور مغزی تا حدودی موفق بوده است، اما هنوز نیاز به بهبود دارد. با تنظیم پارامترها، افزایش داده‌های آموزشی و بهبود معماری مدل، می‌توان به دقت بالاتری در تشخیص تومور دست یافت. این پروژه گام اولیه‌ای در جهت توسعه یک سیستم تشخیص تومور خودکار است و می‌تواند به عنوان پایه‌ای برای تحقیقات آینده مورد استفاده قرار گیرد.



شکل ۶ نمونه هایی از داده های ایجاد شده با استفاده مدل دوم

مدل TriVAE در این تصاویر برای شناسایی تومور از اسلایس‌های MRI استفاده کرده است. حالا بیایید به‌صورت دقیق عملکرد مدل را بررسی کنیم. هر سطر از تصویر شامل چهار نمودار است: اول، اسلایس اصلی (Original Slice)که تصویر MRI ورودی مدل است؛ دوم، خطای بازسازی (Reconstruction Error)که تفاوت بین تصویر بازسازی‌شده و ورودی را نشان می‌دهد و نواحی غیرعادی را مشخص می‌کند؛ سوم، تومور پیش‌بینی‌شده (Predicted Tumor)که خروجی مدل TriVAE است و نواحی تومور را مشخص می‌کند؛ و چهارم، حقیقت زمینی (Ground Truth)که ماسک واقعی تومور است و به عنوان داده مرجع برای ارزیابی عملکرد مدل استفاده می‌شود.

در سطر اول، مربوط به BraTS20_Training_147، مشاهده می‌شود که ماسک پیش‌بینی‌شده مدل نواحی زیادی از مغز را هایلایت کرده، اما بخش‌های اصلی تومور را به‌درستی مشخص نکرده است. مقدار زیادی نویز در پیش‌بینی وجود دارد، زیرا نقاط تصادفی در کل مغز دیده می‌شود و در مقایسه با Ground Truth، مدل موفق نشده است که شکل و اندازه واقعی تومور را به‌درستی شناسایی کند. این نشان‌دهنده دقت پایین و حساسیت بالا ولی کاهش در دقت نواحی شناسایی‌شده است.

در سطر دوم، مربوط به BraTS20_Training_210، مدل برخی از نواحی تومور را شناسایی کرده، اما همچنان بسیاری از بخش‌ها را از دست داده است. میزان نویز نسبت به نمونه اول کمتر شده، اما هنوز بخش‌هایی از مغز که سالم هستند، اشتباهاً به‌عنوان تومور مشخص شده‌اند. در اینجا نیز مدل در تشخیص محدوده‌ی اصلی تومور عملکرد ضعیفی داشته و بخش‌های مهمی را از دست داده است، که نشان‌دهنده حساسیت متوسط، اما کاهش در دقت و پوشش ناقص نواحی توموری است.

با توجه به تصاویر، می‌توان گفت که مدل TriVAE چند مشکل دارد. اول، حساسیت بالا ولی دقت پایین است، زیرا مدل مقدار زیادی از مغز را به‌عنوان ناحیه توموری شناسایی کرده که نشان‌دهنده حساسیت بالا اما دقت پایین است. دوم، خطای بازسازی عمدتاً در مناطقی که تومور وجود دارد، بیشتر دیده می‌شود. این نشان می‌دهد که مدل از Reconstruction Error برای تشخیص استفاده می‌کند، اما در تبدیل آن به ماسک نهایی مشکل دارد. سوم، وجود نویز در ماسک پیش‌بینی‌شده به چشم می‌خورد و برخلاف Ground Truth که دارای مرزهای مشخصی است، خروجی مدل مقدار زیادی نقاط پراکنده و غیرمرتبط دارد.

در نتیجه‌گیری و پیشنهادات برای بهبود مدل، می‌توان به چند نکته اشاره کرد. اول، افزایش کیفیت پس‌پردازش ماسک با استفاده از تکنیک‌هایی مانند morphological operations (erosion, dilation) یا CRF (Conditional Random Fields) می‌تواند نویز را کاهش داده و نواحی پیش‌بینی‌شده را بهبود دهد. دوم، استفاده از داده‌های بیشتری برای آموزش، زیرا ممکن است مدل داده‌های کافی برای یادگیری ویژگی‌های واقعی تومور را دریافت نکرده باشد. استفاده از data augmentation می‌تواند به این امر کمک

کند. سوم، تنظیم آستانه (Thresholding)بهتر، زیرا شاید مدل آستانه‌ای که برای استخراج ماسک نهایی استفاده می‌کند، مناسب نیست. بررسی و بهینه‌سازی این مقدار می‌تواند باعث بهبود نتایج شود. چهارم، افزودن regularization به مدل، به‌ویژه اگر نویز بالا به دلیل overfitting باشد، افزودن dropout، weight decay یا data augmentation می‌تواند مؤثر باشد.

در نهایت، می‌توان گفت که مدل TriVAE عملکرد متوسطی دارد، زیرا در هر دو نمونه مقدار زیادی نویز وجود دارد و مرزهای تومور به‌درستی شناسایی نشده است. مدل در تشخیص کلی برخی از نواحی تومور خوب عمل کرده اما فاقد دقت کافی است. بهینه‌سازی ماسک نهایی و کاهش نویز می‌تواند به بهبود عملکرد آن کمک کند

علل دقت کمتر از مقاله

مدل TriVAE در این تصاویر برای شناسایی تومور از اسلایس‌های MRI استفاده کرده است. با توجه به نتایج به دست آمده و مقایسه با مقاله مرجع، چندین عامل کلیدی می‌تواند باعث کاهش دقت مدل شما نسبت به مقاله مرجع شده باشد. در ادامه، به بررسی هر یک از این عوامل و ارائه تحلیل‌های علمی مربوط به آن‌ها می‌پردازیم.

۱. استفاده از داده‌های دو بعدی به جای سه بعدی

مقاله مرجع: در این مقاله، از آموزش و ارزیابی سه‌بعدی کامل استفاده شده است.

تمرین حاضر: در این پروژه، تنها در بخش‌های اجباری از داده‌های دو بعدی استفاده شده است.

استفاده از داده‌های سه‌بعدی در پردازش تصاویر MRI به مدل این امکان را می‌دهد که ویژگی‌های ساختاری و فضایی تومور را به شکل بهتری استخراج کند. در واقع، تومورها معمولاً در فضای سه‌بعدی دارای ویژگی‌های پیچیده‌ای هستند که تنها با استفاده از تصاویر دو بعدی نمی‌توان آن‌ها را به‌طور کامل شناسایی کرد. مدل‌های دو بعدی ممکن است اطلاعاتی را که در برش‌های دیگر مغز موجود است، نادیده بگیرند، که می‌تواند منجر به کاهش دقت تشخیص و تفکیک تومور شود. به عنوان مثال، تومورهایی که در نواحی خاصی از مغز قرار دارند، ممکن است در یک برش دو بعدی به‌طور کامل قابل مشاهده نباشند، اما در یک حجم سه‌بعدی، ارتباطات و ویژگی‌های آن‌ها به‌وضوح قابل شناسایی است.

استفاده از شبکه‌های سه‌بعدی (۳D CNNs یا VAEهای سه‌بعدی) می‌تواند این مشکل را کاهش دهد و دقت مدل را افزایش دهد.

۲. عدم استفاده از پیش‌پردازش (Skull-Stripping)

مقاله مرجع: در این مقاله، از تکنیک Skull-Stripping برای حذف قسمت‌های غیرضروری مغز استفاده شده است.

تمرین حاضر: در این پروژه، مستقیماً از داده خام استفاده شده است.

تکنیک Skull-Stripping فرآیندی است که قسمت‌های غیرضروری از تصویر مغزی را حذف می‌کند تا مدل فقط روی ناحیه بافت مغزی و تومور تمرکز کند. این فرآیند به کاهش نویز و بهبود دقت تشخیص کمک می‌کند. استفاده از داده‌های خام می‌تواند باعث شود که مدل روی قسمت‌های نامرتبط (استخوان جمجمه، نویز پس‌زمینه) نیز تمرکز کند، که در نتیجه دقت را کاهش می‌دهد. مطالعات نشان داده‌اند که حذف نواحی غیرضروری می‌تواند دقت تشخیص تومور را به‌طور قابل توجهی افزایش دهد.

اضافه کردن یک مرحله پیش‌پردازش برای حذف جمجمه می‌تواند به افزایش دقت مدل کمک کند و باعث شود که مدل تنها بر روی نواحی مرتبط تمرکز کند.

۳. انتخاب پیش‌فرض برای نویز (Coarse + Simplex)

مقاله مرجع: در این مقاله، از Simplex + Coarse استفاده شده است.

تمرین حاضر: در این پروژه، فقط در بخش امتیازی از این ترکیب استفاده شده است.

مدل‌های مبتنی بر VAEمانند TriVAEبه شدت به میزان نویز در داده‌ها وابسته هستند. در مقاله مرجع، مدل با یک ترکیب بهینه از coarse و simplex noise آموزش داده شده است، که می‌تواند تعادل مناسبی بین جزئیات تصویر و کاهش نویز اضافی ایجاد کند. اگر این تکنیک در تمام مراحل استفاده نشده باشد، احتمالاً مدل دچار عدم تعادل در یادگیری ویژگی‌های مهم تومور شده است. به‌علاوه، استفاده از نویز مناسب می‌تواند به مدل کمک کند تا از یادگیری ویژگی‌های غیرضروری جلوگیری کند و به بهبود دقت تشخیص کمک کند.

استفاده کامل و سیستماتیک از coarse و simplex noise در تمام مراحل، نه فقط در بخش امتیازی، می‌تواند به بهبود عملکرد مدل کمک کند.

۴. تعداد کمتر ایپاک (Epochs)

مقاله مرجع: در این مقاله، از بیش از ۵۰ ایپاک استفاده شده است.

تمرین حاضر: در این پروژه، تنها حدود ۲۰ ایپاک اجرا شده است.

مدل‌های عمیق، به‌ویژه VAEها، نیاز به تعداد اپاک‌های بالاتری برای همگرایی دارند. با ۲۰ اپاک، مدل ممکن است هنوز در مرحله Underfitting(عدم یادگیری کافی) باشد. در واقع، تعداد اپاک‌های ناکافی می‌تواند باعث شود که مدل نتواند به‌طور کامل ویژگی‌های موجود در داده‌ها را یاد بگیرد و در نتیجه دقت آن کاهش یابد. در مقاله مرجع، ۵۰+ اپاک استفاده شده است که احتمالاً باعث شده مدل به همگرایی بهتری برسد و دقت آن افزایش یابد.

افزایش تعداد اپاک‌ها و بررسی تغییرات متریک‌ها مانند دقت و از دست دادن اعتبارسنجی (Validation Loss)می‌تواند به بهبود عملکرد مدل کمک کند.

تعریف :Dice Score

Dice Scoreیا (Dice Similarity Coefficient - DSC)معیاری است که برای اندازه‌گیری میزان شباهت بین دو مجموعه دودویی ((Binary Setsاستفاده می‌شود. این معیار به‌ویژه در زمینه‌های پزشکی، مانند سگمنتیشن تصاویر و مقایسه خروجی مدل‌های یادگیری ماشین با Ground Truth، اهمیت زیادی دارد.

فرمول محاسبه Dice Score به شکل زیر است:

$$Dice = \frac{2|A \cap B|}{|A| + |B|}$$

در این فرمول:

Aنمایانگر حجم پیش‌بینی‌شده توسط مدل است.

Bنیز Ground Truthرا نشان می‌دهد.

مقدار Dice Score بین ۰ و ۱ قرار دارد. اگر Dice برابر با ۱ باشد، پیش‌بینی کاملاً برابر با ماسک واقعی است و این دقت ایده‌آل محسوب می‌شود. برعکس، اگر Dice برابر با ۰ باشد، هیچ همپوشانی بین پیش‌بینی و ماسک واقعی وجود ندارد. در عمل، مقدار بالاتر از ۰,۷ معمولاً به‌عنوان یک سگمنتیشن خوب در نظر گرفته می‌شود، هرچند که این مقدار بسته به کاربرد ممکن است متفاوت باشد.

Dice Score در پردازش تصاویر پزشکی اهمیت ویژه‌ای دارد. یکی از مزایای این معیار عدم حساسیت به عدم تعادل کلاس‌ها است. برخلاف دقت (Accuracy)، Dice Score در برابر داده‌های نامتوازن مقاوم است و به‌خوبی میزان هم‌پوشانی را نشان می‌دهد. در مواردی که مدل باید دقیقاً محدوده‌ای را شناسایی کند، Dice یک معیار مناسب است. این معیار به‌طور گسترده در تشخیص تومورها، ضایعات، بافت‌ها و اعضای بدن از تصاویر MRI یا CT اسکن استفاده می‌شود.

در پروژه ما، مقدار Dice Score برای Coarse Noise برابر با ۰,۲۹۶۸ و برای Simplex Noise برابر با ۰,۳۱۶۳ به‌دست آمده است. این نتایج نشان می‌دهند که نویز Simplex توانسته عملکرد مدل را بهبود بخشد. با این حال، هر دو مقدار نسبتاً پایین هستند که نشان‌دهنده نیاز به بهبود مدل، تنظیم آستانه و پردازش بهتر داده‌ها است.

در نتیجه، Dice Score نشان می‌دهد که نویز Simplex توانسته ماسک‌های پیش‌بینی‌شده را کمی بهتر با حقیقت زمین تطابق دهد، اما مدل هنوز جای بهبود دارد. این تحلیل می‌تواند به ما کمک کند تا در مراحل بعدی به بهبود عملکرد مدل و دقت تشخیص بپردازیم.

مقایسه عملکرد VAE با TriVAE

۱. TriVAE

Patient: BraTS20_Training_003, Dice Score: 0.2156

Patient: BraTS20_Training_002, Dice Score: 0.3168

Patient: BraTS20_Training_001, Dice Score: 0.6739

Average Dice score over all patients: 0.2876

۲. VAE

Patient: BraTS20_Training_003, Dice Score: 0.1920

Patient: BraTS20_Training_002, Dice Score: 0.3078

Patient: BraTS20_Training_001, Dice Score: 0.6929

Average Dice score over all patients: 0.2176

مقایسه عملکرد مدل ساده VAE با مدل Tri-VAE نشان‌دهنده تفاوت‌های قابل توجهی در دقت و کارایی این دو مدل در تشخیص تومورهای مغزی است. یکی از معیارهای کلیدی برای ارزیابی عملکرد این مدل‌ها، مقدار Dice Score است که به‌عنوان نمایانگر میزان همپوشانی بین ماسک پیش‌بینی‌شده و ماسک واقعی تومور استفاده می‌شود. هرچه مقدار Dice Score بالاتر باشد، نشان‌دهنده عملکرد بهتر مدل در شناسایی ناحیه تومور است.

نتایج به‌دست‌آمده از ارزیابی مدل‌ها برای بیماران مختلف در دیتاست BraTS20 به‌وضوح تفاوت‌های عملکردی این دو مدل را نشان می‌دهد. به‌طور خاص، مدل Tri-VAE برای بیماران BraTS20_Training_003، BraTS20_Training_002 و BraTS20_Training_001 به ترتیب Dice Scoreهای ۰,۲۱۵۶، ۰,۳۱۶۸ و ۰,۶۷۳۹ را به‌دست آورد که میانگین کلی آن ۰,۲۸۷۶ بود. در مقابل، مدل VAE ساده به ترتیب Dice Scoreهای ۰,۱۹۲۰، ۰,۳۰۷۸ و ۰,۶۹۲۹ را به‌دست آورد و میانگین آن

۰,۲۱۷۶ بود. این مقادیر نشان می‌دهند که مدل Tri-VAE به‌طور کلی عملکرد بهتری نسبت به مدل ساده VAE دارد، به‌طوری‌که میانگین Dice Score مدل Tri-VAE حدود ۷ درصد بالاتر از مدل ساده است.

تحلیل نتایج نشان می‌دهد که مدل Tri-VAE توانسته است درک بهتری از ساختار داده‌ها داشته باشد و اطلاعات بیشتری از فضای نهفته استخراج کند. این بهبود می‌تواند به دلیل طراحی خاص Tri-VAE باشد که به آن اجازه می‌دهد تا ویژگی‌های پیچیده‌تری از داده‌ها را یاد بگیرد. با این حال، تفاوت عملکرد در بیمار BraTS20_Training_001 نشان می‌دهد که مدل ساده VAE در این مورد خاص عملکرد بهتری داشته است. این نکته می‌تواند به این معنا باشد که مدل Tri-VAE هنوز به بهینه‌ترین تنظیمات نرسیده است یا اینکه مدل ساده VAE در برخی موارد به‌خوبی روی الگوهای خاصی آموزش دیده است. بنابراین، این موضوع نیاز به بررسی دقیق‌تری دارد تا مشخص شود آیا مدل Tri-VAE می‌تواند در همه شرایط بهبود یابد یا خیر.

در نهایت، با توجه به نتایج به‌دست‌آمده، می‌توان نتیجه‌گیری کرد که مدل Tri-VAE در مجموع عملکرد بهتری نسبت به VAE ساده دارد. با این حال، بهینه‌سازی‌های بیشتری می‌تواند به تقویت عملکرد آن کمک کند. برای بهبود مدل Tri-VAE، پیشنهاداتی وجود دارد که می‌تواند به افزایش دقت و کارایی آن منجر شود. اول، افزایش تعداد ایپاک‌ها و بررسی همگرایی مدل می‌تواند به یادگیری بهتر و دقیق‌تر ویژگی‌ها کمک کند. این احتمال وجود دارد که مدل هنوز کاملاً همگرا نشده باشد و با افزایش تعداد ایپاک‌ها، بهبود بیشتری در عملکرد آن حاصل شود.

دوم، بررسی تنظیمات فضای نهفته (Latent Space)می‌تواند به افزایش کیفیت بازسازی مدل کمک کند. افزایش تعداد ابعاد فضای نهفته ممکن است به مدل این امکان را بدهد که ویژگی‌های بیشتری را از داده‌ها استخراج کند و در نتیجه دقت تشخیص را افزایش دهد. سوم، تنظیم بهتر وزن‌های تابع هزینه (Loss Function)نیز می‌تواند تأثیر زیادی بر کیفیت نهایی خروجی مدل داشته باشد. ممکن است نسبت KL Divergence به Reconstruction Loss نیاز به تنظیم مجدد داشته باشد تا تعادل بهتری بین این دو عنصر برقرار شود.

در نهایت، استفاده از تکنیک‌های Post-Processing برای بهبود خروجی می‌تواند به کیفیت ماسک خروجی کمک کند. روش‌هایی مانند Conditional Random Fields (CRF) می‌توانند به کاهش نویز و بهبود مرزهای تومور کمک کنند. با این اصلاحات و بهینه‌سازی‌ها، انتظار می‌رود که مدل Tri-VAE عملکرد بهتری نسبت به نتایج فعلی داشته باشد و در نهایت به افزایش کیفیت تشخیص تومورهای مغزی کمک کند. این بهبودها می‌توانند به مدل کمک کنند تا در تشخیص تومورهای مغزی دقت بیشتری داشته باشد و در نهایت به توسعه یک سیستم تشخیص تومور مغزی کارآمدتر منجر شوند.

۱,۴ امتیازی

۱. پردازش سه‌بعدی مدل Tri-VAE

برای ارزیابی بهتر مدل، به‌جای پردازش هر اسلایس به‌صورت جداگانه، خروجی مدل بر روی تمامی اسلایس‌های یک بیمار اعمال شده و به یک حجم سه‌بعدی ترکیب می‌شود. این رویکرد به مدل اجازه می‌دهد تا اطلاعات فضایی بیشتری را در نظر بگیرد و ویژگی‌های ساختاری تومور را به‌طور دقیق‌تری شناسایی کند. به‌ویژه در داده‌های MRI، تومورها معمولاً دارای ویژگی‌های پیچیده‌ای هستند که تنها با استفاده از تصاویر دو بعدی به‌خوبی قابل شناسایی نیستند.

با توجه به محدودیت‌های حافظه، امکان استفاده از هایپرپارامترهای متفاوت وجود دارد که می‌تواند به بهبود عملکرد مدل کمک کند. به‌عنوان مثال، با تنظیم بهینه ابعاد فضای نهفته و تعداد لایه‌های شبکه، می‌توان ویژگی‌های بیشتری از داده‌ها استخراج کرد.

برای بهبود خروجی و حذف نویزهای ناخواسته، فیلتر Median سه‌بعدی بر روی خروجی مدل اعمال می‌شود. این فیلتر به‌ویژه در حذف نویزهای تصادفی و بهبود کیفیت تصویر مؤثر است. همچنین، کامپوننت‌های کوچک که ممکن است ناشی از خطاهای پیش‌بینی یا نویز باشند، حذف می‌شوند تا دقت نهایی مدل افزایش یابد.

در نهایت، شاخص Dice سه‌بعدی بین خروجی مدل و ماسک واقعی محاسبه و گزارش می‌شود. این معیار به‌طور خاص برای ارزیابی دقت مدل در شناسایی نواحی تومور استفاده می‌شود و می‌تواند نشان‌دهنده بهبود عملکرد مدل پس از اعمال پردازش سه‌بعدی باشد.

۲. آزمایش نویز Simplex به‌جای Coarse Noise

در این بخش، مدل Tri-VAE با استفاده از نویز Simplex آموزش داده می‌شود و نتایج آن با مدل قبلی که از Coarse Noise استفاده می‌کرد، مقایسه می‌شود. نویز Simplex یک نوع نویز پیوسته و طبیعی‌تر است که می‌تواند در بهبود عملکرد مدل در تشخیص نواحی آنومالی مؤثر باشد. این نوع نویز به دلیل ساختار پیوسته‌اش، می‌تواند به مدل کمک کند تا ویژگی‌های بیشتری از داده‌ها را یاد بگیرد و از یادگیری الگوهای غیرضروری جلوگیری کند.

مقدار ۳Dice D برای مدل با نویز Simplex محاسبه شده و در یک جدول با مدل Coarse Noise مقایسه می‌شود تا تأثیر هر روش مشخص گردد. این مقایسه به ما این امکان را می‌دهد که به‌طور دقیق‌تری درک کنیم که کدام نوع نویز می‌تواند به بهبود عملکرد مدل کمک کند و آیا تغییر نوع نویز تأثیر معناداری بر دقت تشخیص دارد یا خیر.

هدف این بخش، بررسی تأثیر پردازش سه‌بعدی و نوع نویز بر عملکرد مدل Tri-VAE است. با این تحلیل، می‌توانیم به نتایج بهتری در تشخیص تومورهای مغزی دست یابیم و در نهایت، به توسعه یک سیستم تشخیص خودکار و دقیق‌تر کمک کنیم. این مراحل به‌ویژه در زمینه پزشکی می‌توانند به بهبود کیفیت تشخیص و درمان بیماران مبتلا به تومورهای مغزی منجر شوند و به ارتقاء سطح مراقبت‌های بهداشتی کمک کنند.

نحوه پیاده سازی این دو بخش:

در این بخش، ما دو مرحله مهم را برای بهبود و ارزیابی مدل Tri-VAE انجام دادیم که می‌تواند تأثیر قابل توجهی بر عملکرد آن در تشخیص تومورهای مغزی داشته باشد. اولین مرحله شامل پردازش سه‌بعدی مدل Tri-VAE بود. به‌جای پردازش هر اسلایس به‌صورت جداگانه، ما تصمیم گرفتیم خروجی مدل را بر روی تمامی اسلایس‌های یک بیمار اعمال کرده و آن‌ها را به یک حجم سه‌بعدی ترکیب کنیم. این رویکرد به ما این امکان را می‌دهد که اطلاعات فضایی بیشتری را در نظر بگیریم و ویژگی‌های ساختاری تومور را به‌طور دقیق‌تری شناسایی کنیم.

برای این کار، ابتدا مدل را در حالت ارزیابی قرار دادیم تا از تغییر وزن‌ها و رفتارهای خاصی مانند Dropout جلوگیری کنیم. سپس یک لیست برای ذخیره ماسک‌های پیش‌بینی‌شده‌ی سه‌بعدی ایجاد کردیم. در ادامه، با استفاده از یک حلقه، هر اسلایس دوبعدی را پردازش کردیم. هر اسلایس ابتدا توسط تابع پیش‌پردازش تبدیل به یک تنسور شد و سپس مدل Tri-VAE روی آن اجرا شد. ما بازسازی مدل و نقشه خطای آن را محاسبه کردیم و نقشه خطا را به یک ماسک باینری تبدیل کردیم. این ماسک‌ها در لیست ذخیره شدند و پس از پردازش همه‌ی اسلایس‌ها، خروجی‌ها به‌صورت یک حجم سه‌بعدی ترکیب شدند.

برای بهبود کیفیت این حجم سه‌بعدی، ما از فیلتر Median سه‌بعدی استفاده کردیم تا نویزهای تصادفی را حذف کنیم. همچنین، کامپوننت‌های کوچک که ممکن است ناشی از خطاهای پیش‌بینی یا نویز باشند، شناسایی و حذف شدند. در نهایت، حجم اصلاح‌شده به‌عنوان خروجی تابع برگردانده شد.

پس از آن، ما به محاسبه Dice سه‌بعدی بین ماسک واقعی و ماسک پیش‌بینی‌شده پرداختیم. برای این کار، ماسک‌ها را به‌صورت یک‌بعدی تخت کردیم و مقدار اشتراک بین دو ماسک را محاسبه کردیم. سپس با استفاده از فرمول Dice، دقت تشخیص مدل را ارزیابی کردیم.

در مرحله دوم، ما تصمیم گرفتیم نویز Simplex را به‌جای Coarse Noise امتحان کنیم. برای این کار، تابعی طراحی کردیم که نویز Simplex را روی تصویر اعمال می‌کند. این نویز به‌صورت پیوسته و طبیعی‌تر

نسبت به نویز Coarse است و ما امیدوار بودیم که بتواند عملکرد مدل را در تشخیص نواحی آنومالی بهبود بخشد.

ما یک نسخه جدید از دیتاست سالم ایجاد کردیم و نویز Simplex را به‌عنوان نویز جایگزین برای Coarse Noise استفاده کردیم. سپس یک DataLoader جدید برای این دیتاست ساخته و آن را در TripleLoader مدل جای‌گذاری کردیم.

مدل Tri-VAE جدیدی ساخته شد و به دستگاه منتقل گردید. ما بهینه‌ساز Adam را با نرخ یادگیری e-۳۱ تنظیم کردیم و مدل را به مدت ۲۰ دوره آموزش دادیم. در این مدت، نمودار کاهش خطا را رسم کردیم تا روند یادگیری مدل به‌خوبی مشخص گردد.

پس از آموزش، یک حلقه مشابه با پردازش سه‌بعدی اصلی اجرا کردیم، اما این بار از مدل آموزش‌دیده با نویز Simplex استفاده کردیم. Dice سه‌بعدی برای هر بیمار محاسبه و ذخیره شد و در نهایت، میانگین Dice کل بیماران را چاپ کردیم تا عملکرد مدل با نویز Simplex مشخص گردد.

در پایان، نتایج Dice برای دو مدل، یکی با نویز Coarse و دیگری با نویز Simplex، ذخیره شد. ما یک DataFrame با استفاده از پانداس ساختیم تا مقایسه‌ی دو مدل آسان‌تر شود. سپس جدول نتایج را نمایش دادیم تا تأثیر نویزهای مختلف بر عملکرد مدل مشخص شود.

```
Average 3D Dice over all patients: 0.2968
Average 3D Dice with Simplex Noise: 0.3163
          Method  Average 3D Dice
```

جدول ۱ مقایسه شاخص **Dice** در دو نوع نوییز

Coarse Noise	Simplex Noise
0.2968	0.3163

در تحلیل نتایج خروجی، ما به بررسی عملکرد مدل Tri-VAE با دو نوع نویز مختلف، یعنی Coarse Noise و Simplex Noise پرداختیم. نتایج محاسبه‌شده نشان می‌دهد که میانگین Dice سه‌بعدی برای مدل با Coarse Noise برابر با ۰٫۲۹۶۸ و برای مدل با Simplex Noise برابر با ۰٫۳۱۶۳ است. این نشان می‌دهد که مدل با نویز Simplex عملکرد بهتری داشته و مقدار Dice آن حدود ۲ درصد افزایش یافته است. این بهبود نشان‌دهنده تأثیر مثبت نویز Simplex در بهبود بازسازی مدل است. با این حال، مقدار Dice کمتر از ۰٫۵ نشان می‌دهد که مدل هنوز خطای زیادی دارد و می‌تواند بهبود یابد.

در بررسی اینکه چرا نویز Simplex بهتر عمل کرده است، چندین عامل مهم وجود دارد. اولاً، Coarse Noise به‌عنوان نویزی تصادفی و در مقیاس بزرگ، ممکن است اطلاعات مهمی را از بین ببرد و باعث کاهش دقت مدل شود. در مقابل، نویز Simplex ساختارمندتر است و شباهت بیشتری به نویزهای طبیعی در تصاویر پزشکی دارد. این ویژگی به مدل کمک می‌کند تا بهتر یاد بگیرد که چگونه الگوهای واقعی را حفظ کند، در حالی که Coarse Noise ممکن است به یادگیری نادرست مدل منجر شود

برای بهبود عملکرد مدل، چندین راهکار وجود دارد. یکی از آن‌ها افزایش اندازه دیتاست و استفاده از بیماران بیشتر است تا از بروز پدیده overfitting جلوگیری شود. همچنین، تنظیم هایپرپارامترهای مدل Tri-VAE، مانند نرخ یادگیری و ابعاد فضای نهفته، می‌تواند تأثیر زیادی بر عملکرد مدل داشته باشد. همچنین، افزایش مقدار آستانه threshold=0.1 در مرحله پس‌پردازش می‌تواند به شناسایی نواحی مهم‌تر کمک کند. علاوه بر این، استفاده از تکنیک‌های حذف نویز قوی‌تر، مانند Gaussian Filtering به‌عنوان مکملی برای Median Filter، می‌تواند به بهبود کیفیت خروجی کمک کند

در نهایت، می‌توان نتیجه‌گیری کرد که نویز Simplex باعث بهبود عملکرد مدل شده است، اما هنوز نیاز به بهبود دارد. استفاده از روش‌های پیشرفته‌تر پردازش تصویر و یادگیری عمیق می‌تواند به افزایش مقدار Dice کمک کند. همچنین، آزمایش نویزهای دیگر مانند Perlin Noise یا ترکیب چندین نویز می‌تواند نتایج بهتری به همراه داشته باشد.

پاسخ ۲ – AdvGAN

۲.۱ آشنایی با حملات خصمانه و معماری AdvGAN

مقایسه روش‌های PGD و FGSM با روش AdvGAN و مزیت‌هایAdvGAN

الف) مروری برFGSM (Fast Gradient Sign Method)

۱. تعریف کلی

- روش FGSM یکی از ساده‌ترین روش‌های تولید نمونه‌های خصمانه است که توسط Goodfellow و همکاران (۲۰۱۵) معرفی شد.
- ایده اصلی: از گرادیان تابع هزینه نسبت به ورودی استفاده می‌کند تا با یک گام (Single-step) در جهت علامت گرادیان، تصویر کمی تغییر داده شود.

$$x_{adv} = x + \epsilon \, \text{sign}(\nabla_x J(\theta, x, y))$$

۲. فرمول

- Xتصویر اصلی
- Xadv تصویر خصمانه
- ϵ میزان (یا شدت) اختلال
- ∇xJ(θ,x,y))گرادیان تابع هزینه نسبت به ورودی

۳. مزایا و معایب

- مزیت: بسیار سریع و ساده است (تنها یک گام محاسبه).
- عیب: اختلال معمولاً کوچک است اما ممکن است در برخی مدل‌ها کافی نباشد یا نیاز به تنظیم دقیق ϵ دارد. همچنین در سناریوهایی که مدل مقاوم‌سازی (defense) شده باشد، FGSMممکن است کارایی پایین‌تری داشته باشد.

ب) مروری برPGD (Projected Gradient Descent)

۱. تعریف کلی

- روش PGD نسخه چندگامی (Iterative) از FGSM است و نوعی حمله قوی‌تر محسوب می‌شود.
- در هر گام، در جهت گرادیان حرکت می‌کنیم و سپس نتایج را در یک کره ℓp (معمولاً $\ell\infty$یا $\ell 2$) پیرامون نمونه اصلی پرُروجکت (Projection) می‌کنیم تا میزان اختلال از حد مجاز ϵفراتر نرود.

۲. فرمول ساده‌شده

$$x_{t+1} = \Pi_{x,\epsilon}\{ x_t + \alpha\, \mathrm{sign}(\nabla_{x_t} J(\theta, x_t, y))\}$$

- $\Pi x,\epsilon$ عملگر پرُروجکت کردن روی کره‌ای با شعاع ϵ حول X.
- αگام کوچکتر از ϵ در هر مرحله.

۳. مزایا و معایب

- مزیت: قوی‌تر از FGSM است و اغلب می‌تواند مدل‌های مقاوم‌تر را نیز گول بزند.
- عیب: محاسباتی سنگین‌تر (چند مرحله‌ای) است و در سناریوهای بلک‌باکس ممکن است به سختی قابل اجرا باشد (چون نیازمند گرادیان‌های تکراری است).

پ) مزیت‌های کلی AdvGAN نسبت به PGD وFGSM

۱. تولید نمونه‌های خصمانه به‌صورت مدل مولد (Generator)

- در PGD و FGSM برای ساختن هر نمونه خصمانه، باید مستقیماً از گرادیان و به‌طور تکراری/تک‌مرحله‌ای استفاده کرد.
- در AdvGANیک شبکه مولد (Generator) آموزش داده می‌شود که می‌تواند بدون نیاز به محاسبات گرادیان برای تک‌تک نمونه‌ها، در یک مرحله نمونه خصمانه بسازد. به محض آموزش، تولید نمونه خصمانه بسیار سریع خواهد شد.

۲. انعطاف‌پذیری و تطبیق با سناریوهای مختلف

- در AdvGAN می‌توان با انتخاب توابع هزینه (و قیود مختلف) نمونه‌های خصمانه هدف‌دار (Targeted) یا غیرهدف‌دار (Untargeted) تولید کرد.
- امکان ترکیب توابع هزینه متعدد (برای حفظ کیفیت بصری و همچنین فریب مدل) وجود دارد.

۳. قابلیت تولید انبوه و مقیاس‌پذیر

- چون Generator یک تابع پارامتری آموزش‌دیده است، می‌توان به‌صورت همزمان روی بسیاری از داده‌های ورودی به‌راحتی نمونه خصمانه ساخت.
- برخلاف PGD یا FGSM که هر بار باید محاسبات گرادیانی انجام شود.

تفاوت‌های کلیدی بین AdvGAN و یک GAN ساده

GANساده معمولاً یک مولد (Generator) و یک تمیزدهنده/تفکیک‌کننده (Discriminator) دارد که مولد سعی می‌کند داده‌های جدید شبیه داده‌های واقعی تولید کند و تمیزدهنده تلاش می‌کند واقعی را از مصنوعی تشخیص دهد. در نهایت مولد یاد می‌گیرد داده‌هایی تولید کند که تمیزدهنده نتواند آنها را از داده اصلی تمیز دهد.

در:AdvGAN

۱. هدف مولد

- مولد (Generator) به‌جای تولید داده‌های «واقعی‌نما»، نمونه‌هایی تولید می‌کند که به یک کلاس اشتباه در مدل هدف (Target model) منجر شوند یا احتمال پیش‌بینی یک کلاس خاص را بالا ببرند (حملهٔ هدف‌دار یا غیرهدف‌دار).

۲. نقش تمیزدهنده (Discriminator)

- در یک GAN ساده، تمیزدهنده سعی می‌کند واقعی بودن نمونه‌ها را تشخیص دهد؛ ولی درAdvGAN ، تمیزدهنده می‌تواند معیاری برای بررسی «تشابه بصری نمونه خصمانه با نمونه اصلی» باشد (به‌طور مثال با تفکیک «نمونه اصلی» از «نمونه دستکاری‌شده»).
- همچنین از خروجی مدل هدف (یا تابع هزینه مربوط) در کنار تمیزدهنده استفاده می‌شود.

۳. توابع هزینه

- در GAN ساده، تابع هزینه مولد همان فاصله بین توزیع داده‌های حقیقی و مصنوعی است.
- در AdvGAN توابع هزینه می‌توانند چندبخشی باشند :

۱. خطای فریب (دقتاً مرتبط با خروجی مدل هدف که باید گول بخورد)

۲. فاصله یا شباهت بین تصویر خصمانه و تصویر اصلی (تا پنهان ماندن تغییرها)،

۳. هزینه‌های مبتنی بر تمیزدهنده برای نگه داشتن کیفیت ظاهری نمونه یا جلوگیری از تشخیص آسان دستکاری

---

۳. نحوهٔ استفاده AdvGAN از خروجی یا گرادیان‌های مدل هدف در زمان آموزش

- چرا به خروجی مدل هدف نیاز داریم؟

  برای این که بدانیم مولد در تولید نمونه خصمانه موفق عمل کرده یا نه، باید از پاسخ مدل هدف (Target Model)یا گرادیان آن استفاده کنیم. در حالت وایت‌باکس(White-box) ، گرادیان مستقیماً در دسترس است و می‌توانیم مانندFGSM ، PGDو... از آن استفاده کنیم. در حالت بلک‌باکس، ممکن است تابع هزینه را با پرس‌وجوی خروجی مدل تخمین بزنیم.

- روش کلی در AdvGAN

۱. Generatorیک تصویر کاندید (خصمانه) ایجاد می‌کند.

۲. این تصویر به مدل هدف داده می‌شود.

۳. خروجی مدل هدف (احتمال کلاس‌ها یا لاگیت‌ها) و/یا گرادیان‌های آن محاسبه می‌شوند.

۴. تابع هزینه بر اساس انحراف از پاسخ مطلوب (مثلاً اجبار به اشتباه کلاس) و همچنین حفظ شباهت با ورودی اصلی محاسبه و به Generator بازگردانده می‌شود.

۵. Generatorبا استفاده از این سیگنال خطا به‌روزرسانی پارامتر را انجام می‌دهد.

---

۴. نحوهٔ تولید نمونه‌های خصمانه در AdvGAN و حفظ «فداکاری-کیفیت» حمله

۱. تولید از طریق Generator

- در هر حلقهٔ آموزش، Generatorورودی اصلی (مثلاً تصویر اصلی) را گرفته و یک اختلال (perturbation)یا نسخهٔ تغییریافته از آن را خروجی می‌دهد.
- این خروجی باید هم کلاس مورد نظر (یا هدف دلخواه) را در مدل هدف فعال کند، و هم از نظر بصری شباهت بالایی به تصویر اصلی داشته باشد.

۲. مفهوم «فداکاری» یا «سازش» (Trade-off)

- در روش‌های خصمانه، همیشه یک سازش بین «قابلیت فریب مدل» و «کیفیت بصری یا شباهت با تصویر اصلی» وجود دارد.
- AdvGANمعمولاً در تابع هزینه‌اش بخشی را به نگه داشتن تغییرات کوچک‌تر در پیکسل‌ها اختصاص می‌دهد (مثلاً استفاده از(ℓ2یا normℓ∞)و بخشی دیگر را به ماکسیمم کردن خطای مدل هدف.
- ازاین‌رو باید پارامترهایی برای وزن‌دهی به این دو جنبه تعیین کرد.

5.بررسی سه تابع هزینه اصلی در مقاله AdvGAN و نقش هر کدام

در اکثر پیاده‌سازی‌های مقاله AdvGAN (و مقالات مرتبط) سه جزء هزینه اصلی در نظر گرفته می‌شوند (البته بسته به نسخه مقاله ممکن است تعداد یا نام‌های متفاوتی داشته باشند، اما رایج‌ترین‌ها عبارت‌اند از):

۱. هزینهٔ فریب مدل هدف (Adversarial Loss)

- این بخش تلاش می‌کند خروجی مدل هدف را به کلاس دلخواه (حمله هدف‌دار) یا اشتباه (حمله غیرهدف‌دار) سوق دهد.
- اگر مدل هدف را fدر نظر بگیریم، و برچسب مطلوب ما y^*باشد، هزینهٔ فریب می‌تواند به شکل $CE(f(x_{adv}), y^*)$یا عبارت‌های مبتنی بر لاگیت باشد.

۲. هزینهٔ شباهت با تصویر اصلی (Reconstruction/Similarity Loss)

- برای این که تغییرات ایجاد شده در تصویر محسوس نباشد، اغلب از یک هزینهٔ فاصله (مثلاً ℓ2یا ℓ∞ بین تصویر اصلی xو نمونه خصمانه xadvاستفاده می‌شود.
- هرچه این فاصله کمتر باشد، تصویر خروجی طبیعی‌تر به نظر می‌رسد.

۳. هزینهٔ تمیزدهنده یا متمایزکننده (Discriminator Loss)

- همانند GAN عادی، تمیزدهنده سعی می‌کند بین تصویر «واقعی» و «تغییریافته» تمایز قائل شود . Generatorمی‌خواهد این تمایزدهنده را گول بزند تا نمونه خصمانه، «طبیعی» جلوه کند.
- ترکیبی از این هزینه با هزینه فریب مدل هدف باعث می‌شود هم کیفیت بصری حفظ شود و هم مدل هدف دچار خطا شود.

چگونه این سه بخش با هم ترکیب می‌شوند؟

:معمولاً تابع هزینه کلی به‌صورت جمع وزنی از این سه مؤلفه است؛ مثلاً

$$L_{total} = \lambda_1 \cdot L_{adversarial} + \lambda_2 \cdot L_{similarity} + \lambda_3 \cdot L_{discriminator}$$

مقادیر λiبر اساس اولویت و اهمیت هر بخش تنظیم می‌شوند.

6.جمع‌بندی

- روش‌های FGSMو PGDبا تکیه بر به‌روزرسانی‌های گرادیانی مستقیماً نمونه خصمانه می‌سازند، اما تولید پیوسته و انبوه نمونه‌ها در آنها هزینه‌بر است.
- AdvGANبا اضافه کردن معماری GAN بر ایدهٔ حملات خصمانه، یک Generator آموزش می‌دهد که می‌تواند به‌سرعت نمونه خصمانه تولید کند؛ همچنین با درنظرگرفتن توابع هزینه مختلف، کنترل کیفیت بصری و میزان فریب امکان‌پذیر می‌شود.
- تفاوت‌های کلیدی AdvGAN و GAN ساده در هدف و معیارهای ارزیابی (فریب مدل هدف و حفظ شباهت تصویری) و جایگاه تمیزدهنده نهفته است.
- در نهایت سه مؤلفه هزینهٔ مهم در مقاله AdvGAN عبارت‌اند از :هزینهٔ فریب)برای فریب مدل هدف(، هزینهٔ شباهت (برای حفظ ویژگی‌های ظاهری) و هزینهٔ تمیزدهنده) (Discriminator) برای بهبود کیفیت بصری و جلوگیری از تشخیص آسان خصمانه بودن)

۲,۲ پیاده سازی مدل AdvGAN

۱.مقدمه

در دهه‌های اخیر، یادگیری عمیق به عنوان یکی از پیشرفت‌های بزرگ در حوزه هوش مصنوعی شناخته شده است. با این حال، این مدل‌ها در برابر حملات adversarial آسیب‌پذیر هستند؛ به عبارت دیگر، با اضافه کردن نویزهای کوچک به داده‌های ورودی، می‌توانند به اشتباه بیفتند و نتایج نادرستی ارائه دهند. این پروژه به بررسی تأثیر حملات adversarial بر روی مدل‌های یادگیری عمیق با استفاده از مجموعه داده CIFAR-10 می‌پردازد و تلاش می‌کند تا با استفاده از روش‌های مختلف، مقاومت مدل‌ها را در برابر این حملات ارزیابی کند.

۲.اهداف پروژه

۱. آشنایی با مجموعه داده :CIFAR-10 دانلود و آماده‌سازی داده‌ها برای آموزش و ارزیابی مدل‌ها.
۲. پیاده‌سازی مدل‌های یادگیری عمیق :استفاده از مدل‌های از پیش آموزش‌دیده شده مانند ResNet20.
۳. اجرای حملات :adversarial استفاده از روش‌های مختلف مانند FGSM و شبکه‌های GAN برای تولید نمونه‌های.adversarial
۴. ارزیابی و تحلیل نتایج :بررسی دقت مدل‌ها در مقابل داده‌های اصلی و adversarial و تحلیل میزان موفقیت حملات.

۳.مراحل اجرای پروژه

۱ .نصب کتابخانه‌های مورد نیاز : در ابتدا، کتابخانه‌های TensorFlow و CleverHans برای اجرای حملات adversarial نصب شدند.

۲ .وارد کردن کتابخانه‌ها : کتابخانه‌های ضروری مانندPyTorch ، TensorFlow، Matplotlibو دیگر ابزارهای مورد نیاز برای پردازش داده‌ها و مدل‌سازی وارد شدند.

۳ .آماده‌سازی داده‌ها :با استفاده از تبدیل‌های مختلف، داده‌های CIFAR-10 دانلود و به مجموعه‌های آموزشی، اعتبارسنجی و تست تقسیم شدند.

۴ .تقسیم‌بندی داده‌ها :مجموعه داده به سه بخش آموزش، اعتبارسنجی و تست تقسیم شد.

خروجی:

Files already downloaded and verified

Files already downloaded and verified

۵ .نمایش نمونه‌ای از داده‌ها :برای بررسی کیفیت داده‌ها و درک بهتر آن‌ها، تعدادی از تصاویر تصادفی نمایش داده شدند.

خروجی :

شکل ۷نمونه‌ای از تصاویر مجموعه آموزشی **CIFAR-10**

۶ .بارگذاری و ارزیابی مدلResNet20

مدل ResNet20 از PyTorch Hub بارگذاری شد و دقت آن بر روی مجموعه تست ارزیابی گردید.

خروجی:

ResNet-20 model achieved an accuracy of 92.46% on the CIFAR-10 test dataset.

۷ .تنظیم محیط محاسباتی

تشخیص دستگاه موجود CPU یا GPU و تنظیم TensorFlow برای استفاده بهینه از منابع.GPU

خروجی:

Computation will be performed on: cpu

۸ .تولید نمونه‌های Adversarial با استفاده ازFGSM

روش Fast Gradient Sign Method (FGSM) برای تولید نمونه‌های adversarial بر روی داده‌های تست استفاده شد.

۹ .ارزیابی نمونه‌هایAdversarial

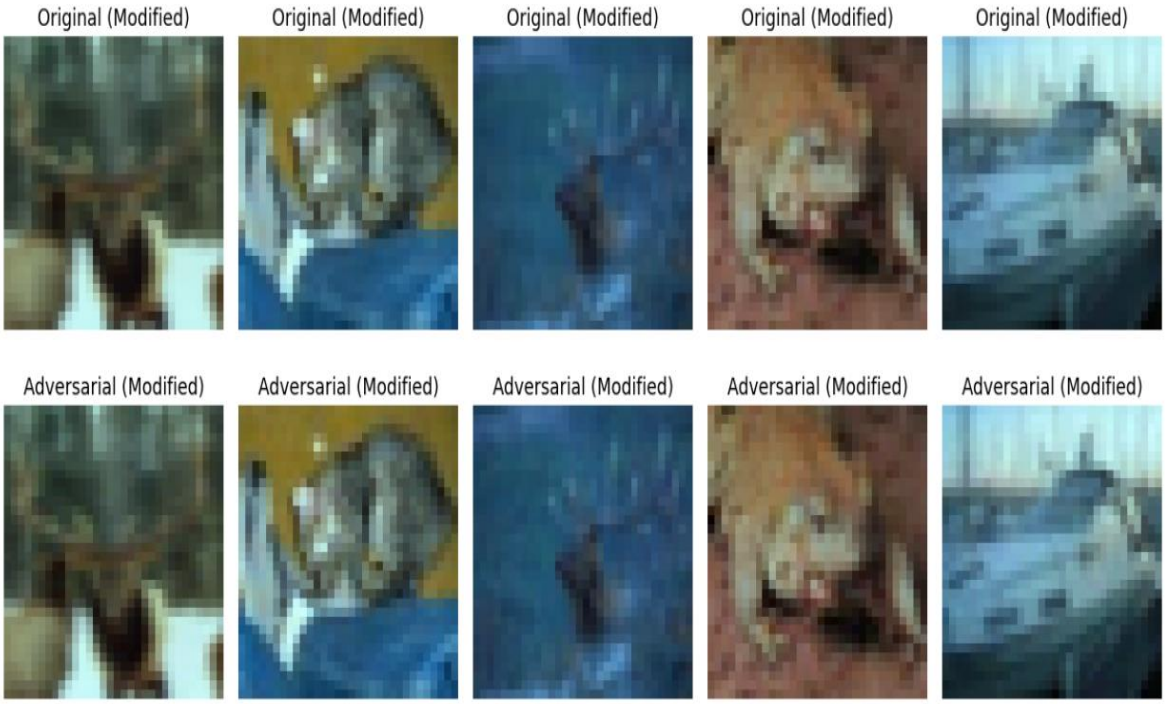
مدل ResNet20 بر روی نمونه‌های adversarial اجرا شد تا دقت آن‌ها ارزیابی گردد.

خروجی:

tensor([3, 8, 5, 3, 3, 5, 3, 5, 5, 3, 5, 9, 5, 7, 8, 3, 5, 3, 8, 3, 5, 0, 5, 9,5, 5, 5, 5, 3, 3, 5, 5])

Using cache found in C:\Users\98930/.cache\torch\hub\chenyaofo_pytorch-cifar-models_master

۱۰ .نمایش تصاویرAdversarial



شکل ۸نمایش تصاویر اصلی و **adversarial**

۱۱ .آموزش شبکه‌های GAN برای تولید نمونه‌هایAdversarial

شبکه‌های Generator و Discriminator برای تولید و تشخیص نمونه‌های adversarial پیاده‌سازی و آموزش داده شدند.

۱۲ .اجرای فرآیند آموزشGAN

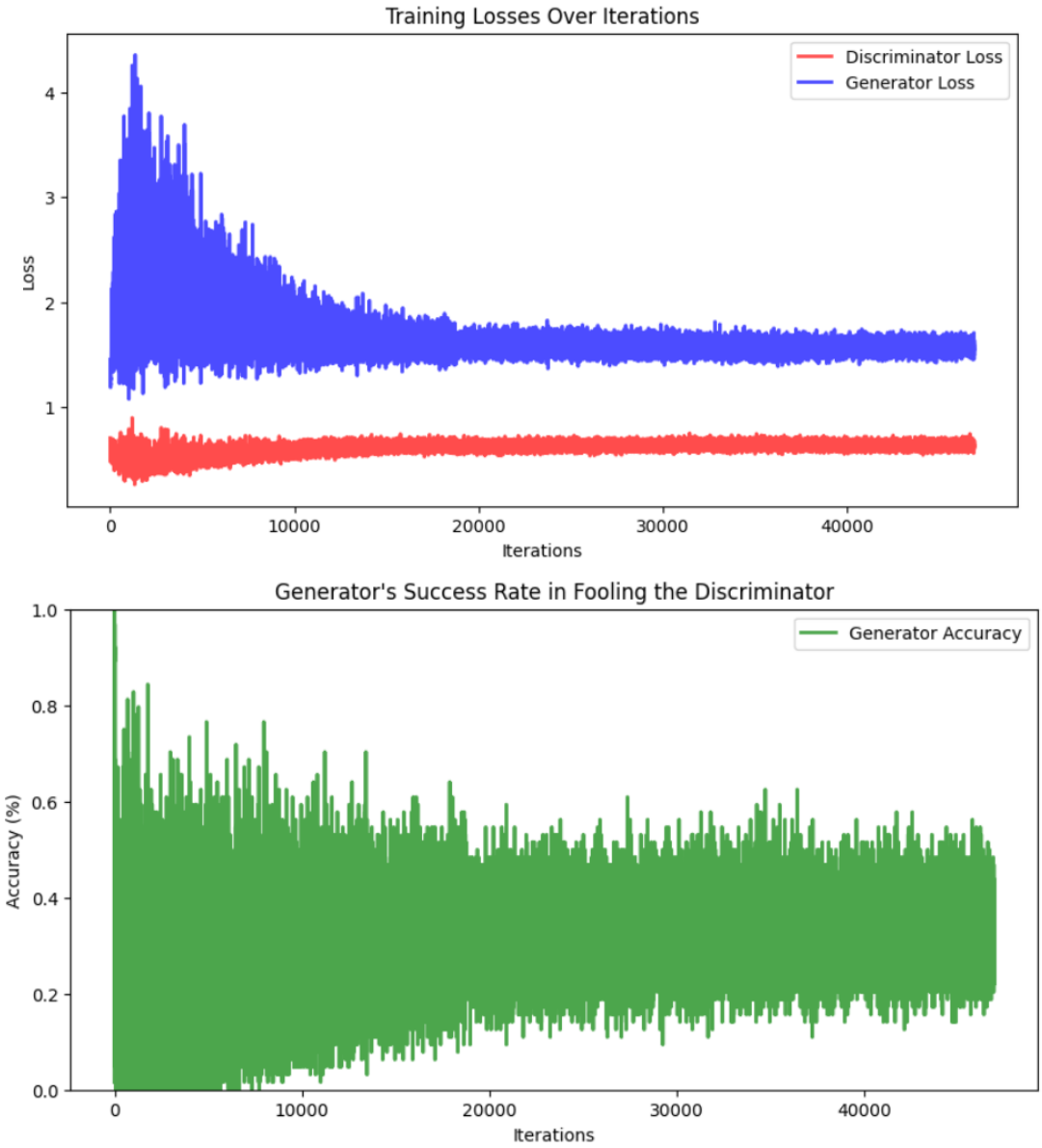
شبکه‌های Generator و Discriminator به مدت ۵۰ دوره آموزشی آموزش یافتند. نمودارهای تغییرات loss و دقت در طول آموزش به شرح زیر است:

خروجی:

Epoch 1/50, Loss D: 0.5053, Loss G: 2.0561, Acc: 0.1157

...

Epoch 50/50, Loss D: 0.6375, Loss G: 1.5773, Acc: 0.3322



شکل ۹نمودار تغییرات **Loss** و دقت در طول دوره‌های آموزشی

۱۳ .ارزیابی حملاتAdversarial

موفقیت حملات adversarial به صورت کلی و بر اساس هر کلاس محاسبه شد.

خروجی:

Overall success rate of the adversarial attack: 88.19%

Class 0: Attack Success Rate = 96.80%, Accuracy = 92.60%

Class 1: Attack Success Rate = 100.00%, Accuracy = 96.80%

Class 2: Attack Success Rate = 100.00%, Accuracy = 90.80%

Class 3: Attack Success Rate = 71.30%, Accuracy = 85.10%

Class 4: Attack Success Rate = 100.00%, Accuracy = 93.10%

Class 5: Attack Success Rate = 14.80%, Accuracy = 88.50%

Class 6: Attack Success Rate = 99.50%, Accuracy = 95.00%

Class 7: Attack Success Rate = 100.00%, Accuracy = 93.10%

Class 8: Attack Success Rate = 99.50%, Accuracy = 95.30%

Class 9: Attack Success Rate = 100.00%, Accuracy = 94.30%

Classification Report (Original Images):

	precision	recall	f1-score	support
0	0.9232	0.9260	0.9246	1000
1	0.9613	0.9680	0.9646	1000
2	0.8919	0.9080	0.8999	1000
3	0.8527	0.8510	0.8519	1000
4	0.9101	0.9310	0.9204	1000
5	0.8939	0.8850	0.8894	1000
6	0.9341	0.9500	0.9420	1000
7	0.9678	0.9310	0.9490	1000
8	0.9578	0.9530	0.9554	1000
9	0.9554	0.9430	0.9492	1000
accuracy			0.9246	10000
macro avg	0.9248	0.9246	0.9246	10000
weighted avg	0.9248	0.9246	0.9246	10000

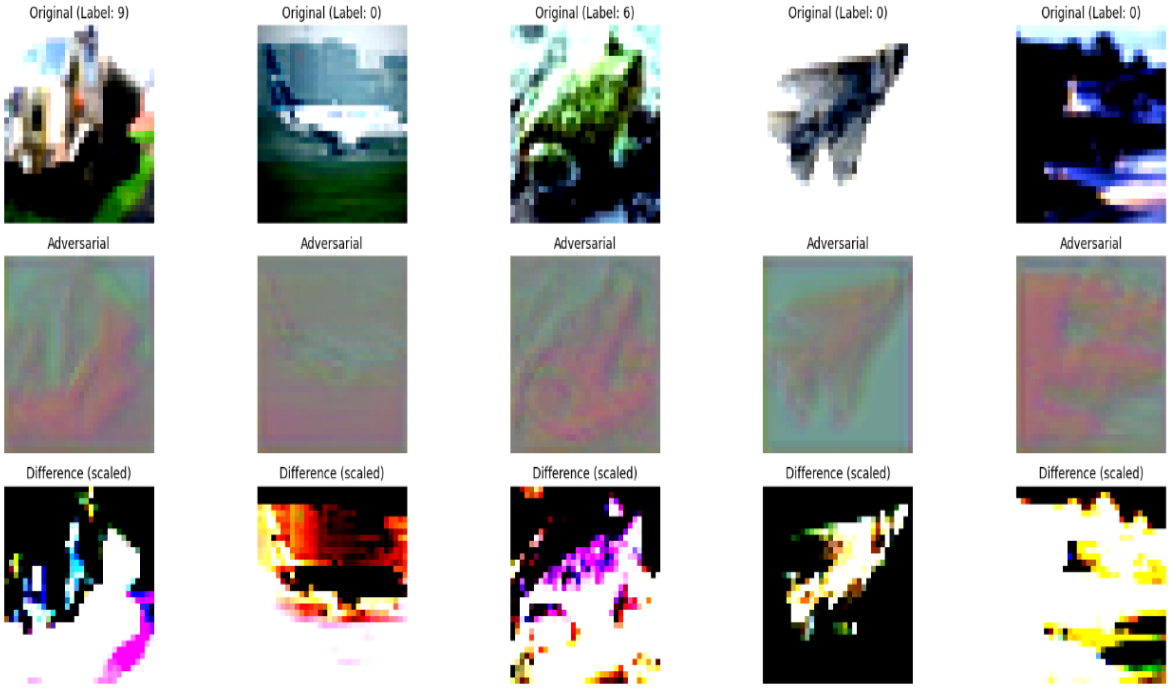
Classification Report (Adversarial Images):

	precision	recall	f1-score	support
0	0.1481	0.0320	0.0526	1000
1	0.0000	0.0000	0.0000	1000
2	0.0000	0.0000	0.0000	1000
3	0.0795	0.2870	0.1246	1000
4	0.0000	0.0000	0.0000	1000
5	0.1406	0.8520	0.2414	1000

6	0.0538	0.0050	0.0091	1000
7	0.0000	0.0000	0.0000	1000
8	0.2083	0.0050	0.0098	1000
9	0.0000	0.0000	0.0000	1000
accuracy		0.1181	10000	
macro avg	0.0630	0.1181	0.0438	10000
weighted avg	0.0630	0.1181	0.0438	10000

۱۴ .نمایش نمونه‌های Adversarial و تفاوت‌های آن‌ها

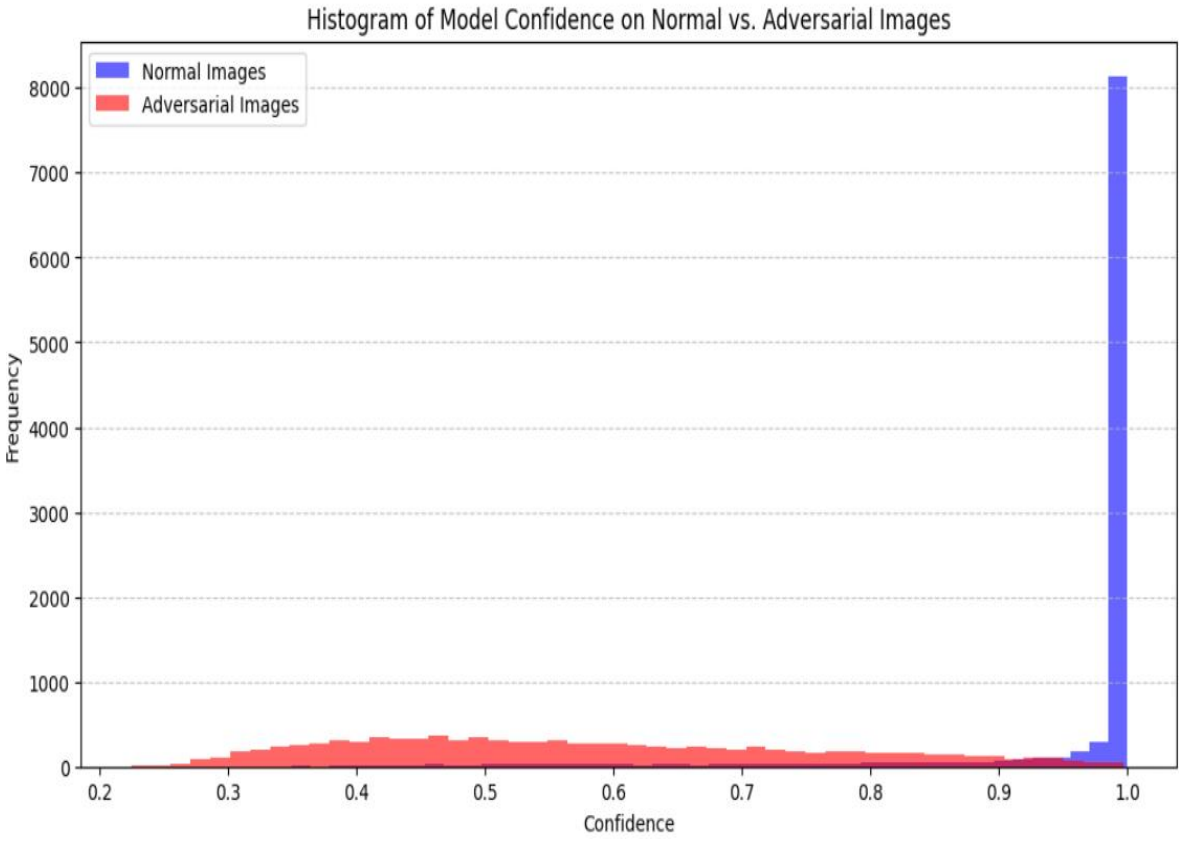
تصاویر اصلی، adversarialو تفاوت‌های آن‌ها برای ۵ نمونه به صورت زیر نمایش داده شدند:



شکل ۱۰نمایش تصاویر اصلی، **adversarial** و تفاوت‌های آن‌ها

۱۵ .نمایش هیستوگرام اطمینان مدل

هیستوگرام اطمینان مدل در برابر تصاویر اصلی و adversarial نمایش داده شد.



شکل ۱۱هیستوگرام اطمینان مدل بر روی تصاویر اصلی و **adversarial**

۴.تحلیل نتایج

۱. دقت مدل بر روی داده‌های اصلی :مدل ResNet20 با دقت ۹۲,۴۶٪ در دسته‌بندی صحیح تصاویر مجموعه تست عمل کرد که نشان‌دهنده عملکرد قوی آن در شناسایی دسته‌های مختلف است.

۲. تأثیر حملات :adversarial نرخ موفقیت حملات adversarial به طور کلی ۸۸,۱۹٪ بود که نشان‌دهنده آسیب‌پذیری مدل در برابر چنین حملاتی است. برخی کلاس‌ها مانند کلاس‌های ۱، ۲، ۴، ۷، ۸ و ۹ نرخ موفقیت ۱۰۰٪ داشتند که نشان‌دهنده حساسیت بالا این کلاس‌ها به حملات adversarialاست.

۳. تحلیل دسته‌بندی :مدل در تصاویر اصلی دقت بالایی داشته ولی در تصاویر adversarial دقت آن به شدت کاهش یافته است. این مسئله نشان‌دهنده نیاز به توسعه مدل‌های مقاوم‌تر در برابر حملات adversarial است.

۴. بصری‌سازی تفاوت‌ها :تفاوت‌های قابل توجهی بین تصاویر اصلی و adversarial مشاهده شد که نشان‌دهنده تغییرات کوچکی است که می‌تواند تأثیر بزرگی بر تصمیمات مدل داشته باشد.

۵.نتیجه‌گیری

پروژه حاضر نشان داد که مدل‌های یادگیری عمیق مانند ResNet20 در برابر حملات adversarial آسیب‌پذیر هستند. با استفاده از روش‌های مختلف تولید نمونه‌های adversarial و تحلیل‌های دقیق، توانستیم میزان تأثیر این حملات را بر عملکرد مدل ارزیابی کنیم. این نتایج اهمیت توسعه و بهبود روش‌های مقاوم‌سازی مدل‌ها در برابر حملات adversarial را برجسته می‌سازد.

۶.پیشنهادات برای تحقیقات آتی

۱. بهینه‌سازی مدل‌ها :توسعه مدل‌های مقاوم‌تر با استفاده از تکنیک‌های regularization و adversarial training.
۲. استفاده از روش‌های حمله پیشرفته‌تر :بررسی تأثیر روش‌های مختلف حمله adversarial بر روی مدل‌ها.
۳. گسترش مجموعه داده‌ها :استفاده از مجموعه داده‌های بزرگ‌تر و متنوع‌تر برای ارزیابی بهتر مقاومت مدل‌ها.
۴. تحلیل بیشتر بر اساس ویژگی‌ها :بررسی تأثیر ویژگی‌های خاص تصاویر بر روی موفقیت حملات adversarial.

توضیحات اضافی برای دستیار محترم

- لیست شکل‌ها :در این گزارش، پنج شکل شامل نمونه تصاویر، نمایشadversarial ، نمودارهای آموزش، تفاوت تصاویر و هیستوگرام اطمینان مدل وجود دارد.
- کپشن‌ها :هر تصویر با یک کپشن توضیحی همراه است تا به درک بهتر نتایج کمک کند.

- امتیازی:

۱- برای شروع، داده‌های CIFAR-10 را بارگذاری و پیش‌پردازش کردیم. این داده‌ها به صورت تنسورهای نرمال‌سازی شده تبدیل شدند و مجموعه‌های آموزشی و آزمایشی برای آموزش و ارزیابی مدل‌ها آماده شدند. سپس، مدل‌های مورد نیاز را تعریف کردیم. ما یک مدل تولیدکننده

(Generator)طراحی کردیم که وظیفه‌اش تولید تصاویر خصمانه بود و همچنین یک مدل هدف (Target Model)برای شناسایی این تصاویر.

۲- در مرحله بعد، تابعی برای ایجاد حملات خصمانه هدفمند طراحی کردیم که به طور خاص به یک کلاس هدف، در اینجا کلاس ۱، حمله می‌کند. در این تابع، از گرادیان‌های پیش‌بینی‌ها برای ایجاد تغییرات در تصاویر استفاده شد. پس از تولید تصاویر خصمانه، نرخ موفقیت حمله را محاسبه کردیم. نرخ موفقیت به عنوان نسبت تصاویری که به درستی به کلاس هدف شناسایی شدند، تعیین شد و نتایج نشان داد که برای کلاس هدف ۱، نرخ موفقیت برابر با ۸۹,۸۱٪ به دست آمد. این نشان می‌دهد که ۸۹,۸۱ درصد از تصاویر خصمانه به درستی به عنوان کلاس ۱ شناسایی شده‌اند و این نرخ موفقیت بالا نشان‌دهنده اثرگذاری حملات خصمانه بر روی مدل‌های یادگیری عمیق است.

۳- در ادامه، نتایج را به صورت بصری نمایش دادیم. پنج تصویر از مجموعه آزمایشی به همراه تصاویر خصمانه و تفاوت‌های بین آنها به نمایش درآمدند. این تصاویر به وضوح نشان می‌دهند که چگونه تغییرات جزئی در تصاویر می‌تواند منجر به تغییر در پیش‌بینی‌های مدل شود. همچنین، هیستوگرام قطعیت مدل برای تصاویر عادی و خصمانه ترسیم شد تا تفاوت‌های موجود در پیش‌بینی‌ها را نشان دهد.

۴- این پروژه نشان داد که حملات خصمانه هدفمند می‌توانند به طور قابل توجهی بر روی دقت مدل‌های یادگیری عمیق تأثیر بگذارند. نرخ موفقیت ۸۹,۸۱٪ نشان‌دهنده توانایی بالای حملات خصمانه در فریب مدل‌ها است. این نتایج می‌تواند به بهبود روش‌های دفاعی در برابر حملات خصمانه کمک کند و نیاز به تحقیق بیشتر در این زمینه را نشان می‌دهد.

تفاوت مدل های Targetedو :Untargeted

در حملات هدفمند، هدف این است که یک تصویر به گونه‌ای تغییر داده شود که مدل یادگیری عمیق به اشتباه آن را به یک کلاس خاص شناسایی کند. به عبارت دیگر، در این نوع حمله، مهاجم تعیین می‌کند که تصویر باید به کدام کلاس خاص هدایت شود. برای مثال، اگر تصویر یک گربه باشد، مهاجم ممکن است بخواهد که مدل آن را به عنوان یک سگ شناسایی کند. در اینجا، هدف این است که مدل به طور خاص به اشتباه یک پیش‌بینی مشخص را انجام دهد.

این نوع حملات معمولاً با استفاده از گرادیان‌های تابع هزینه طراحی می‌شوند، به طوری که تغییرات ایجاد شده در تصویر، بیشترین تأثیر را بر روی پیش‌بینی کلاس هدف داشته باشد. حملات هدفمند می‌توانند به طور خاص در سناریوهای امنیتی و کاربردهای حساس مانند شناسایی چهره و تشخیص اشیاء خطرناک بسیار مهم باشند، زیرا می‌توانند منجر به اشتباهات جدی در تصمیم‌گیری‌های خودکار شوند.در مقابل،

حملات غیرهدفمند به گونه‌ای طراحی شده‌اند که هدف آن‌ها فقط ایجاد اشتباه در پیش‌بینی مدل است، بدون اینکه به یک کلاس خاص اشاره کنند. به عبارت دیگر، در این نوع حمله، مهاجم فقط می‌خواهد که مدل به اشتباه یک تصویر را به هر کلاسی غیر از کلاس واقعی آن شناسایی کند. برای مثال، اگر تصویر یک گربه باشد، هدف این است که مدل آن را به هر کلاسی غیر از "گربه" شناسایی کند، مثل "سگ"، "پرنده" یا "ماشین".

حملات غیرهدفمند معمولاً ساده‌تر از حملات هدفمند هستند و می‌توانند با استفاده از تکنیک‌هایی مانند Fast Gradient Sign Method (FGSM) یا Projected Gradient Descent (PGD) پیاده‌سازی شوند. این نوع حملات بیشتر بر روی کاهش دقت کلی مدل تمرکز دارند و می‌توانند به عنوان یک روش عمومی برای ارزیابی پایداری مدل‌ها در برابر حملات خصمانه استفاده شوند.